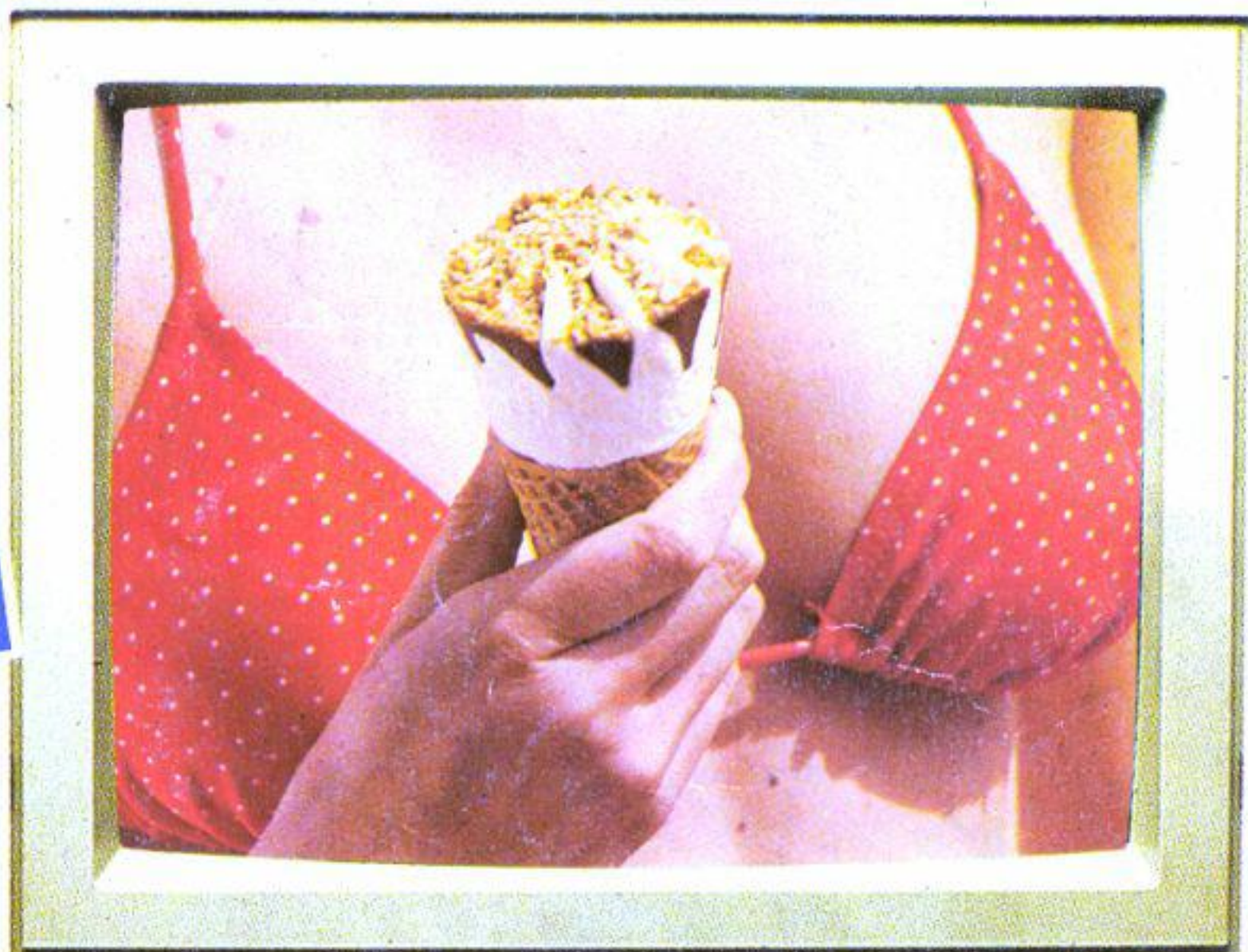
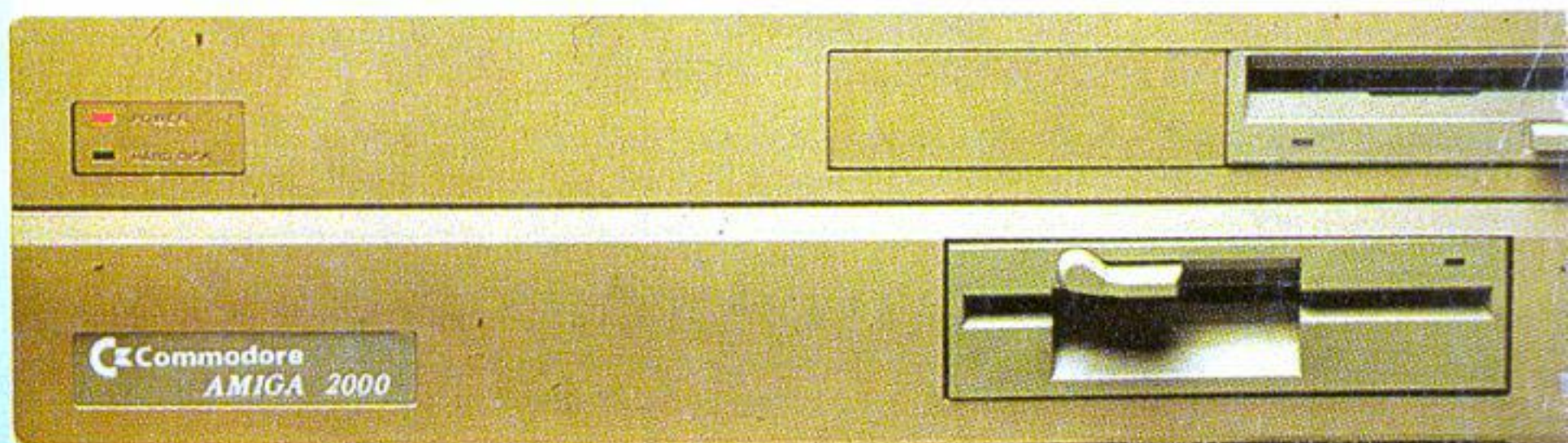
25 luglio/25 agosto 1987 - Anno VI - N° 44 - Sped. Abb. Post. Gr. III/70 - CR - Distr. MePe

Amiga 500 e 2000 all'orizzonte

SIP, è di modem il 64

Suoni e musica col 128

Peek & Poke per C/16

Ora anche su disco

MS-DOS
&
GW-BASIC
EMULATOR

TRASFORMA
IL TUO 64
IN UN
PC IBM

"MS-DOS & GW-BASIC emulator" è anche su disco. Per quanti hanno acquistato la versione su cassetta ed inviano la relativa prova d'acquisto, il dischetto è disponibile a lire 15.000 (+ lire 3.000 per spese di spedizione). Non occorre inviare la cassetta nè tantomeno il manualetto di istruzioni. Chi non è in possesso della cassetta può richiedere il disco ed il manualetto al prezzo normale di lire 25.000 (+ lire 3.000 per spese di spedizione).

Per una veloce evasione dell'ordine inviate un assegno bancario o circolare non trasferibile all'ordine della "Systems Editoriale" (V.le Famagosta, 75 - 20142 Milano).

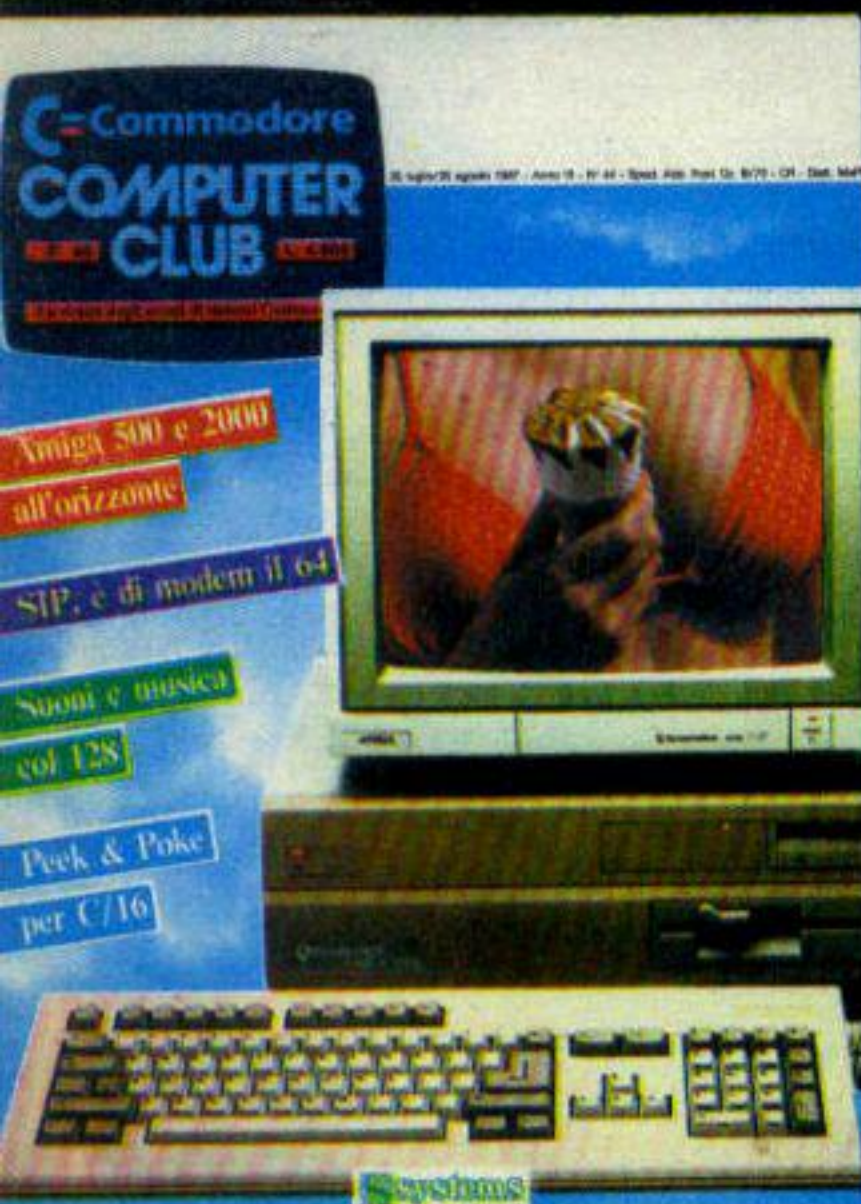
Sempre un passo avanti.

Sommario

INSERTO

VIAGGIO ALL'INTERNO DEL BASIC

RUBRICHE

Copertina: L'immagine che appare nel monitor è tratta dal film Sanson realizzato dalla BRW per l'agenzia Bjke.

Direttore: Alessandro de Simone - **Caporedattore:** Michele Maggi
Redazione/collaboratori: Paolo Agostini, Claudio Baiocchi, Carlo e Lorenzo Barazzetta, Simone Bettola, Luigi Callegari, Sergio Camici, Sandro Certi, Umberto Colapicchioni, Maurizio Dell'Abate, Valerio Ferri, Giancarlo Mariani, Roberto Marigo, Clizio Merli, Marco Miotti, Roberto Morassi, Antonio Pastorelli, Carla Rampi, Marco Saetta, Fabio Sorgato, Danilo Toma, Giovanni Valli.
Segretaria di redazione: Maura Ceccaroli
Ufficio Grafico: Arturo Ciaglia
Direzione, redazione, pubblicità: v.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Milano: Leandro Nencioni (direttore vendite), Guido Agosti, Giorgio Ruffoni, Claudio Tidone - v.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
• Emilia Romagna: Spazio E - P.zza Roosevelt, 4 - 40123 Bologna - Tel. 051/236979
• Toscana, Marche, Umbria: Mercurio srl - via Rodari, 9 - San Giovanni Valdarno (Ar) - Tel. 055/947444
• Lazio, Campania: Spazio Nuovo - via P. Foscari, 70 - 00139 Roma - Tel. 06/8109679
Segreteria: Marina Vantini - **Abbonamenti:** Liliana Spina
Tariffe: prezzo per copia L. 4.000. Abbonamento annuo (11 fascicoli) L. 40.000. Estero: il doppio. Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 80.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizione: Systems Editoriale Srl - **Fotolito:** Systems Editoriale Srl
Stampa: La Litografica Srl - Busto Arsizio (Va)
Registrazioni: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70% - **Distrib.:** MePe - via G. Carcano, 32 - Milano

Ricomincio da Amiga

Buone notizie dalla Commodore che sembra impegnata, più che mai, a conquistare una consistente fetta di mercato

Da quando l'ingegner Assi ha preso il "comando" della Commodore Italiana, molte cose sono cambiate e molte altre, assicurano, cambieranno presto.

Ne è una prova lo spirito di iniziativa che ha portato a termine numerose operazioni di un certo impegno.

Il nuovo staff tecnico è costituito da elementi validi e, soprattutto, competenti e consci di ciò che fanno. Con uno di loro ci siamo intrattenuti un intero pomeriggio, in Redazione, a parlare di Amiga, C/64, interfacce, periferiche, software, hardware, tecniche di programmazione e iniziative commerciali; un pomeriggio, insomma, trascorso tra veri "appassionati", che non ci saremmo mai sognati di trascorrere qualche tempo fa, prima del cambio della guardia.

L'epoca del "vendi ora, al dopo ci pensa Iddio" sembra esser definitivamente tramontata; inizia, al contrario, la fase dell'assistenza degna del prestigio di cui, giustamente, godono gli stupendi computer Commodore.

Il modem con software su Rom non è altro che la prima di altre proposte per il C/64, computer che, hanno dichiarato ufficialmente, sarà prodotto e sostenuto per altri anni ancora.

L'Amiga, ovviamente, rappresenterà il fiore all'occhiello che consentirà di presentarsi a testa alta nel mercato dei personal computer. L'Ms-Dos, infatti, deve esser considerato obsoleto, soprattutto dopo la presentazione ufficiale dei nuovi modelli dell'IBM, che sembra insistere nella pretesa di dettar legge nella fascia media dei PC (ma il tentativo di un PC junior IBM fallì miseramente).

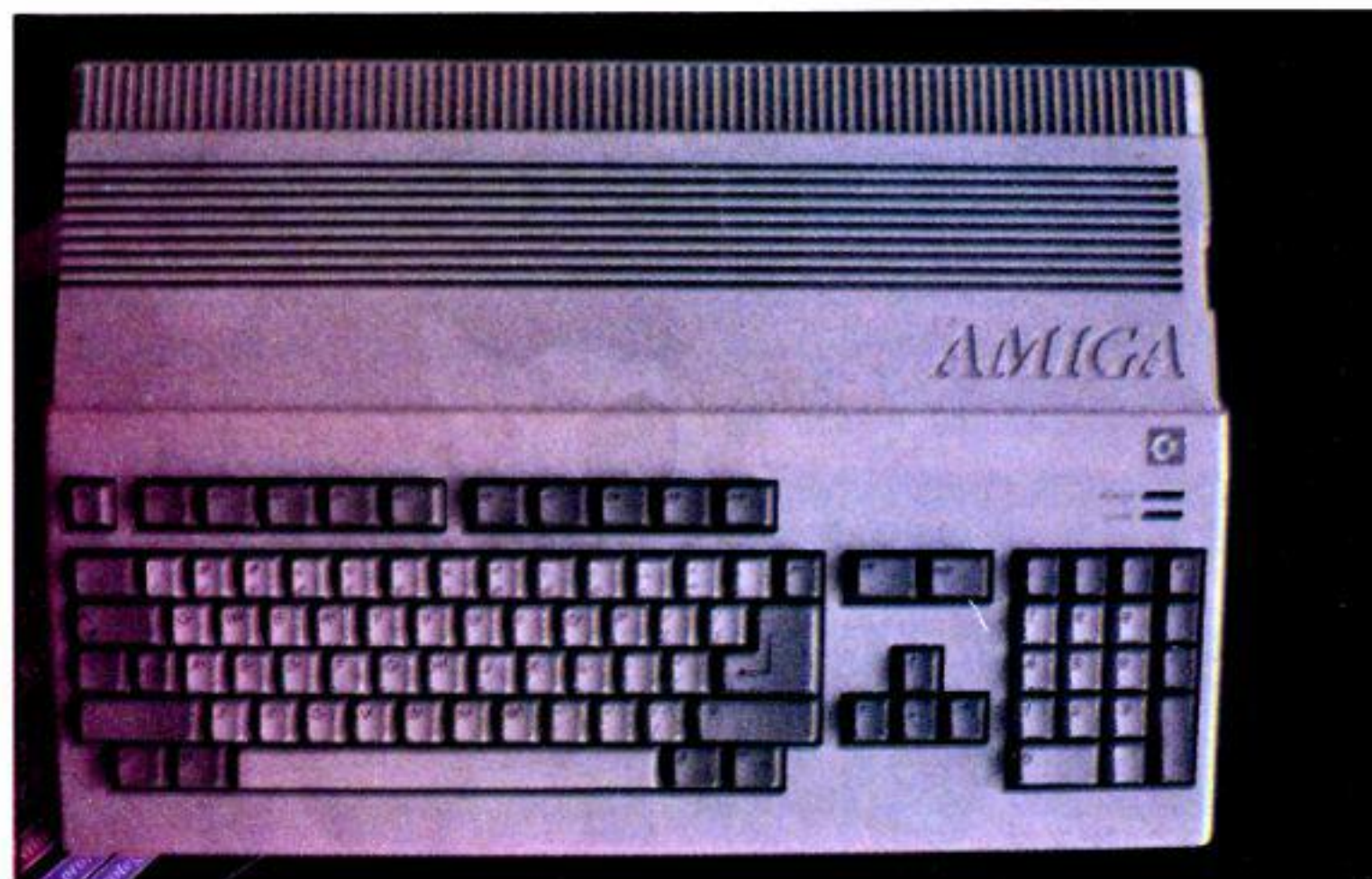
Con l'Amiga, del resto, eventuali desideri Ms-Dossiani dovrebbero esser soddisfatti a breve termine, grazie alla scheda hardware che consentirebbe la totale compatibilità Ms-Dos.

Il nuovo software specifico per Amiga non dovrebbe aver nulla da invidiare, comunque, allo standard Ms-Dos erchè molti programmi svolgeranno tutte le funzioni dei più noti package IBM e, magari, qualcuna in più.

Naturalmente noi siamo qui a guardare, a provare e, soprattutto, a riferire.

Ma è indubbio che, almeno osservando ciò che finora è stato fatto dal nuovo staff della Commodore Italiana, le promesse sono state mantenute.

E, con i tempi che corrono, dobbiamo apprezzare un simile atteggiamento in chi vuole iniziare.

Anzi, ricominciare...

Alessandro de Simone

Amiga incompatibile

☐ Perchè alcuni programmi commercializzati che girano perfettamente con il Workbench 1.1 non funzionano se si inserisce la nuova release 1.2?
(da alcune telefonate)

• Quando la Commodore divulgò, presso le varie software house che ne fecero richiesta, notizie particolareggiate sulla versione 1.1 del sistema operativo, tenne a sottolineare la necessità di rispettare alcuni "Entry point" del sistema stesso. In pratica, a livello di linguaggio macchina, la Commodore garantiva la futura compatibilità del software che avesse rispettato la corretta gestione di alcuni indirizzi assoluti.

Non tutte le software house, però, rispettarono le regole, e a qualcuna venne in mente di gestire grafica e suono utilizzando non gli indirizzi suggeriti, ma altri, che sembravano meglio rispettare le esigenze del software in sviluppo.

La conseguenza consiste, dunque, nel fatto che i programmi in oggetto funzionano benissimo a patto di caricare prima la versione 1.1 (di workbench e Kickstart); con la release 1.2 si manifestano gli inconvenienti lamentati dei quali non può essere incolpata la Commodore, ma le ditte del settore che hanno sottovalutato l'importanza delle notizie diffuse.

Un archivio codificato

☐ Come è possibile, nel programma "Archivio codificato per ricerche And/Or" (CCC N.37) fare in modo di individuare i record interessati nelle ricerche incrociate?
(Francesco Quagliarello - Milano)

• E' sufficiente modificare la riga 31500 come segue:

```
31500 Print#4,"Elemento n."
i-1;Ar$(i,x1)
```

Il numero i-1, infatti, rappresenta l'elemento visualizzato in quel momento; in seguito, tornato al menu principale, puoi rintracciarlo con l'opzione "Esame elementi" oppure "Correzione" per apportare eventuali modifiche.

Errori

☐ Nella risposta data ad un lettore, ho notato due errori per ciò che riguarda la lettura della directory e l'esistenza di espansioni di memoria per il C/64.
(Davide Granzotto - Conegliano Tv)

• Ammetto di aver toppato. La sintassi corretta per la lettura della directory di un drive 3040 è infatti...

```
Load "$0",8
```

...e non...

```
Load "0:$",8
```

...come asserito sul numero di Commodore Computer Club incriminato.

Se, però, non merito perdono per l'errata (involontaria) segnalazione, non sono d'accordo con l'arguto lettore per ciò che riguarda l'espansione di memoria del C/64.

Anche noi leggiamo le riviste americane, e posso assicurare, anzi, che la parte che leggo più volentieri è proprio la pubblicità sia per rendermi conto delle novità offerte, sia per tenermi aggiornato sull'evoluzione dei "gusti" del consumatore.

So benissimo che esistono espansioni Ram (anche di marca Commodore), ma la mia perplessità nel parlarne è dovuta al fatto che preferisco divulgare notizie realmente utili ai nostri lettori.

Non mi interessa lo "scoop" in quanto tale: per noi sarebbe facilissimo rintracciare qualcuno, tra i nostri conoscenti, che si reca spesso negli Stati Uniti, incaricarlo di procurarsi uno di questi meravigliosi apparecchietti e dedicare qualche pagina sulla rivista. Ma a chi servirebbe?

Solo se vi fosse un numero accettabile di programmi che utilizzano tali prodotti hardware ne parleremmo volentieri.

In caso contrario preferiamo dedicare lo spazio della nostra rivista ad argomenti che possano accontentare realmente la maggior parte dei lettori.

Consulenze offronsi

☐ Ho realizzato un accessorio harware, e relativo software, che...
Ho scritto un programma che...
Potrebbe interessare?
(Mauro Ghirelli, Fabio Fontana, Marco Milletti, Davide Granzotto e altri lettori).

• Abbiamo più volte detto che la collaborazione dei lettori è gradita. E' opportuno, però, che il primo contatto sia telefonico e non epistolare, proprio per stabilire, in tempo reale, se le offerte interessano o meno.

Se, infatti, ci chiedete se può interessare un programma di musica (o che riguardi un qualsiasi altro settore), la risposta è sempre e certamente positiva.

Ma il passaggio dalla teoria (=proposta) alla pratica (=redazione dell'articolo e del programma) non sempre è facile e siamo stati costretti, nel passato, a rifiutare la collaborazione di lettori che proponevano listati rivelatitisi, poi, pieni di errori, di incongruenze o altre amenità. Analogamente sono stati cestinati listati decisamente interessanti e versatili, ma privi di un adeguato sostegno... giornalistico (vale a dire privi di articolo che spiegasse il perchè e il percome).

Per ciò che riguarda l'hardware preferiamo pubblicare schemi di SEMPLICISSIMI accessori, SEMPRE corredati, oltrè che dall'articolo, anche da programmi dimostrativi. Nel caso la realizzazione sia complessa, l'autore dovrebbe impegnarsi, almeno moralmente, a vendere per corripondenza il kit di costruzione,

oppure l'apparecchio già pronto e funzionante.

Per informazioni sulla collaborazione, comunque, ripetiamo ciò che abbiamo detto alcune decine di volte: telefonate (tel.02/84.67.34.8) PRIMA di inviare listati e/o articoli.

Il buffer Rs-232

☐ In quale porzione di memoria sono allocati i 512 byte che vengono gestiti dall'interfaccia Rs-232?
(Nicola D'ingegno - Roma)

• Non appena si dichiara l'apertura del file relativo all'interfaccia Rs-232 viene effettuata, automaticamente, un'istruzione Clr che cancella tutte le variabili dichiarate fino a quel momento. Il sistema riserva infatti, per le necessarie manipolazioni, due banchi contigui da 256 byte ciascuno. Il primo, individuabile dalla coppia di puntatori $00F7, è il buffer ricevente; l'altro (puntato da $00F9) gestisce la trasmissione dei dati.

E' quindi indispensabile, in un programma Basic, che l'istruzione Open sia la prima del programma stesso e, ovviamente, che Close sia l'ultima: anche quando si chiude il canale Rs-232, infatti, viene effettuato un Clr automatico con la conseguente perdita di tutte le varialbili Basic fino a quel momento elaborate.

Si sconsiglia, comunque, la gestione di un file Rs-232 per mezzo del Basic, soprattutto operando ad alte velocità. Nel caso della ricezione, infatti, le varie attese per la garbage collection, per l'interpretazione dei comandi e, magari, per la contemporanea memorizzazione su disco, fa perdere istanti (e caratteri) preziosi.

L'ideale è un programma in linguaggio macchina che provveda a memorizzare i dati in ricezione in una zona Ram ben protetta. Al termine della stessa ricezione si potrà, con tutta calma, provvedere a riversare su supporto magnetico i dati accumulati.

Forse è così

☐ Sono un ragazzo di 14 anni e ritengo di essere in grado di spiegare il Bug di Easy Script segnalato sul N.41 (art. "Sillabatore").

Easy Script, secondo me, non appena individua, all'interno di una parola, il primo trattino-codice idoneo per la divisione in sillabe, interrompe subito la ricerca e non verifica l'eventuale presenza di altri trattini all'interno della stessa parola, ma passa subito ad esaminare quella successiva.

Si potrebbe ovviare inserendo un trattino non ad ogni sillaba (come proposto nel programma pubblicato) ma, circa, a metà parola...
(Fabio Andolfi - Givoletto)

• ...e sperare che la fortuna faccia capitare ogni fine riga con la metà selezionata! Scherzi a parte è molto probabile che la diagnosi sia esatta e mi congratulo per la perspicacia dimostrata.

Per ciò che riguarda il programma modificato, che hai spedito solo su carta, non ci è possibile, purtroppo, trovare tempo per digitarlo, verificarlo e (quindi) pubblicarlo.

Ma se decidi di inviarci altri lavori (stavolta su disco) ricordati di aggiungerlo: lo pubblicheremo volentieri su "Directory", a patto che ti ricordi di inserire, all'interno dello stesso programma, istruzioni per l'uso sufficientemente dettagliate.

Leggere le avvertenze

☐ Sono un vostro assiduo lettore e vorrei sapere come posso fare per proteggere i file di un disco da cancellazioni involontarie.
(Paolo Ferri - Verona)

• Una tirata d'orecchi al nostro non-lettore: il programma che consente di fare ciò che viene chiesto, infatti, è stato di recente pubblicato sul n.36 (art: "File recuperato mezzo salvato"), fascicolo con numerosi articoli dedicati al 1541.

Ne approfitto per ricordare ai lettori che molto spesso alcuni listati pubblicati contengono numerose opzioni la cui utilità non sempre può essere "riassunta" nel titolo. L'articolo in oggetto, tanto per fare un esempio, avremmo dovuto intitolarlo:

"Come recuperare un file accidentalmente cancellato, come proteggere (e sproteggere) un file da involontarie cancellazioni, come cambiare il tipo di file, come individuare il codice dei tipi di file; e altre informazioni..."

Bisognerebbe prendere l'abitudine di leggere ciascun articolo della rivista, anche se, in apparenza, sembra non destare il vostro interesse: molto spesso alcune informazioni, che cercate disperatamente per mari e per monti, sono buttate lì, tra un commento ad una Rem e la riflessione sull'uso di alcune Poke...

Posizioni particolari

☐ Perchè, quando registro un programma su disco, non sempre il suo nome compare per ultimo, ma occupa una posizione che sembra casuale?
(Antonio De Giorgi - Marciano di Leuca)

• Quando un disco viene formattato, la zona di memoria riservata alla directory viene cancellata ed inizializzata opportunamente in modo da poter trascrivere (a patto di avere spazio a sufficienza), fino ad un massimo di 144 file diversi. In altre parole è come se il drive avesse a disposizione "solo" 144 righe in cui arrangiarsi per realizzare l'indice (la directory, appunto) che servirà come punto di riferimento per la memorizzazione dei dati su disco.

Con il seguente programma, infatti, ti accorgi che non è possibile superare il valore 144 (pena la comparsa del messaggio "Disk Full"), nonostante siano disponibili, al momento della segnalazione dell'errore, oltre 500 blocchi liberi sul disco.

```
100 for i=1 to 145
110 a$=str$(i)
120 save a$,8
130 print a$;: next
```

Se, in un disco appena formattato, registri tre programmi dal nome, rispettivamente, di "Primo", "Secondo" e "Terzo", chiamando successivamente la directory ti accorgerai che, in effetti, sono stati registrati l'uno dopo l'altro.

Supponiamo, ora, di cancellare il programma "Secondo". Nella directory i due nomi "Primo" e "Terzo"

compaiono ancora l'uno sotto l'altro, ma ciò è dovuto al software di gestione dei nomi: nella realtà, invece, c'è un "buco" in corrispondenza delle locazioni in cui era prima presente il nome (e le altre informazioni) relative al programma "Secondo".

Se, a questo punto, registri un programma dal nome "Quarto", il sistema operativo del drive lo allocherà nel primo spazio disponibile che, appunto, è quello rimasto vuoto dopo la cancellazione di "Secondo".

Dovrebbe essere chiaro, quindi, che se registri più file in successione, senza cancellarne alcuno, questi vengono memorizzati nello stesso ordine in cui sono stati salvati; se, invece, ne cancelli qualcuno, il prossimo file che registrerai occuperà la prima zona lasciata libera.

C'è da dire che non sempre ciò è vero perchè, talvolta, se lo spazio lasciato libero è troppo piccolo, il sistema operativo provvede egualmente a memorizzare il nuovo file in coda a quelli già presenti sul disco.

Non dovrebbe comunque esser difficile realizzare un programma che, leggendo la directory, provveda a riordinare, secondo le esigenze dell'utente, tutti i nomi ivi presenti. Sono molti i programmi professionali di utility per drive che contengono questa comoda (?) opzione.

Quanti colori?

□ Ho saputo che il Commodore 16 dispone di ben 121 colori. Il mio C/64 è limitato ai soli 16 colori, oppure è possibile ottenerne un numero maggiore?
(Vito Catania - Catania)

• Il C/16 (e il Plus-4) dispone di 16 colori (numerati da 1 a 16); per ciascuno di essi è possibile selezionare uno degli otto livelli di luminosità (numerati da 0 a 7).

Il C/16 dovrebbe disporre di 128 tonalità diverse (16*8=128) ma, in pratica, quelle del bianco coincidono, anche se al contrario, con quelle del nero ed i "colori" (termine improprio) risultano, in totale, 121.

Non è invece possibile variare la luminosità dei colori del C/64, se non ricorrendo a tecniche particolari che

provvedono a cambiare rapidamente il colore di (piccole) zone di memoria video con lo scopo di visualizzare miscugli di colori diversi.

Ma è compatibile?

□ In seguito all'acquisto del drive, mi sono procurato una cartuccia speciale con la speranza di riversare su disco tutto il software che possedevo su nastro. Pur se la cartuccia funziona sul C/64 di un mio amico, non funziona su almeno l'80% dei programmi che tento di riversare sul mio C/128 in modo 64. Perchè?
(Giuseppe Prestigiacomo casella postale 556 - Palermo)

• Non penso che il mancato funzionamento della cartuccia dipenda dalla incompatibilità del C/128 con il "vero" C/64, ma ritengo che le cause siano da attribuire ad altri fattori difficilmente diagnosticabili in base alle scarse notizie fornite.

Pubblichiamo comunque, come richiesto nella lettera, il recapito del lettore nella speranza che qualcuno che abbia avuto (e risolto) il problema sollevato possa dare una mano allo sfortunato 128ista.

Profittatori

□ Perchè il prezzo del software che proponete su disco è circa il doppio di quello su nastro dal momento che voi stessi affermate che un disco costa, grossomodo, quanto un nastro?
(Sandro Misuri - Firenze)

• Ti assicuro che ci farebbe molto piacere mettere in vendita i dischi a prezzo inferiore: il numero di acquirenti, infatti, aumenterebbe di molto e compenserebbe ampiamente il minor ricavo ottenibile per ciascuna confezione.

Ti assicuro, peraltro, che non approfittiamo del fatto che i possessori di un drive sembrano essere più "ricchi" di coloro che posseggono il solo datassette (come insinua un altro lettore...).

Il solo costo di riproduzione di un disco (a parte, quindi, il materiale e la confezione) è infatti superiore al costo del materiale, della riproduzione e della confezione di una cassetta. Il motivo, apparentemente strano, di una simile affermazione è dovuto a ragioni di mercato: le ditte che riproducono cassette sono in numero maggiore (per ora) di quelle che riproducono dischi e sono quindi costrette a limitare i guadagni per ovvie ragioni.

Quando coloro che riproducono dischi supereranno un certo numero, il costo (e quindi il prezzo) scenderà di conseguenza.

Tieni presente che fino a poco tempo fa non era nemmeno pensabile la proposta di dischi ai nostri lettori, a causa dei notevoli prezzi richiesti dalle rare ditte del settore che offrivano il servizio di duplicazione. Se queste ultime sono scese a più miti consigli è dovuto non tanto ad un abbassamento del costo dei dischi, ma solo al fatto che il mercato si dirige rapidamente verso la saturazione e le pretese devono necessariamente essere ridimensionate...

Anzi, ti dirò di più: i prezzi delle apparecchiature per la duplicazione sono scesi in modo incredibile ed è probabile che noi stessi ne acquisteremo una, evitando di far ricorso a Ditte esterne.

Syntax error

□ Ho digitato con la massima cura il listato "Alla scoperta delle tracce" (CCC N.40) ma, dato il Run, si blocca con la segnalazione di "Syntax Error in 120". Dov'è l'errore, dal momento che la riga è identica a quella riportata sulla rivista?
(Sandro Saba - Villacidro)

• Sicuramente in un errore di trascrizione da parte tua, e mi permetto di suggerire una diagnosi: una "O" invece dello zero (0); una "I" invece di un "1" e simili. Oppure ad una errata trascrizine degli argomenti delle Poke che, agendo erroneamente sull'area Basic, hanno combinato un gran pasticcio.

Se mi avessi inviato il listato su output di stampante (o meglio, su disco) sarei stato certamente più preciso.

Controllo di un VCR

□ E' possibile controllare un videoregistratore per mezzo della User Port del mio C/64?
(Vincenzo Camporeale - Margerita di S.)

• La User Port è uno dei due connettori a pettine "orizzontali" posti sul retro del C/64. Insieme ai vari Pin ivi presenti, sono disponibili otto "denti" del pettine che, opportunamente programmati con una manciata di righe Basic, possono presentare alla propria estremità una tensione di +5 Volt (oppure 0 Volt); questa, mediante opportuni componenti (transistor, relè e simili) può gestire otto interruttori collegati ad altrettanti apparecchi oppure sostituire otto interruttori di uno strumento che, nel caso specifico, può essere il telecomando del VCR (o, magari, di un TV).

Come puoi intuire, però, è necessario non solo disporre dell'apposito apparecchio da connettere alla User Port, ma anche manomettere il telecomando (o la tastiera del VCR) per effettuare i vari collegamenti necessari; e, come al solito, è opportuno che l'operazione sia affidata a qualcuno più che esperto di montaggi elettronici per evitare disastrose conseguenze.

Cercatemi l'errore

□ Ho scritto un programma da me ideato, di cui allego l'output di stampante, ma non gira correttamente; sapreste dirmi che errore ho commesso?
(Paolo Sciascia - Palermo)

• Semplice: non ci hai inviato il programma su disco! Scherzi a parte, il tempo a nostra disposizione è limitatissimo (come forse abbiamo detto alcuni decenni addietro) e non ci è assolutamente possibile trovarne per effettuare consulenze del genere.

E' sicuro che, prima o poi, affronteremo l'argomento relativo alla tecnica di programmazione usata (e mal applicata, a quanto sembra) nel programma inviato. Nel frattempo, auguri!

Il grande software made-in-Italy

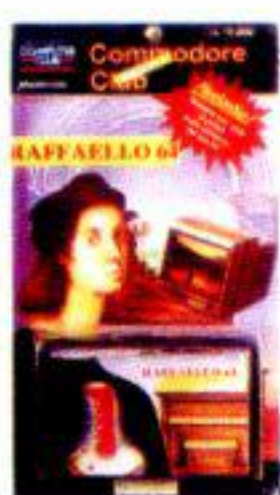
LA VOCE III

Fa parlare e cantare il C64 secondo come lo programmi senza l'uso di campionatori nè sintetizzatori.
Tutte le parole o le canzoni così prodotte possono essere inserite come stringhe in altri programmi.

Lire 12.000

RAFFAELLO

Un programma per disegnare col tuo Comodore 64 col solo joystick senza Koala nè tavoletta grafica.
Tutti i disegni prodotti possono essere memorizzati ed uti lizzati in altri programmi.

Lire 10.000

OROSCOPO

Fa in maniera scientifica l'oroscopo personale. Il più completo programma astrologico per Commodore 64.

Lire 12.000

COMPUTER MUSIC

Un music-editor avanzato più per un programma juke-box con 27 motivi celebri di musica classica e leggera da Arcadia a Bach, Vivaldi, Zeppelin...

Lire 12.000

GESTIONE FAMILIARE

Tre programmi su cassetta che giustificano l'aggettivo "domestico" del tuo computer:
- bilancio familiare;
- dieta equilibrata;
- scheda medica familiare.

Gira su C/64/128

Lire 12.000

BANCA DATI

Un potente data base per C/64 e Spectrum disponibile anche su disco.
L'edizione su cassetta contiene da un lato la versione C64 e dall'altro la versione Spectrum.

Lire 12.000

DICHIARAZIONE DEI REDDITI (740/S)

Programma aggiornato al 1986 per la dichiarazione dei redditi, modello semplificato. Per C64.

Disco: **Lire 24.000**
Cassetta: **Lire 16.000**

MATEMATICA FINANZIARIA

Pubblicato a puntate su Commodore (n.ri 13, 14 e 15) e su Personal Computer (n.ri 1, 2, 3 e 4) questo programma offre un vero e proprio corso completo di ragioneria su Commodore 64.
Se ne consiglia l'acquisto insieme agli arretrati delle riviste che ne illustrano l'uso ed il funzionamento.

Disco: **lire 20.000**
Cassetta: **lire 10.000**

Comodore 13, 14 e 15 e Personal Computer 1, 2, 3
Lire 21.000

ANALISI DI BILANCIO

Naturale completamento di "Matematica Finanziaria" questo programma consente di calcolare automaticamente tutti i ratio più significativi e di confrontare due bilanci dello stesso ente.
Il testo esplicativo è stato pubblicato su Personal Computer n.ri 2, 3, 4 e 5 che si consiglia di acquistare contemporaneamente.

Disco: **Lire 20.000**
Cassetta: **Lire 10.000**

Personal Computer 2, 3, e 5: **Lire 12.000**

ARREDARE

Un programma professionale per ottimizzare le soluzioni d'arredamento della vostra casa.
N.B. gira solo sotto Simon's Basic.

Disco: **Lire 20.000**
Cassetta: **Lire 10.000**

GRAPHIC EXPANDER 128

Un potente programma grafico per il c 128 in modo 128.

Lire 27.000

Sì, inviatemi i seguenti software al prezzo di listino + Lire 3.000 per spese di spedizione:

- ☐ RAFFAELLO
- ☐ LA VOCE III
- ☐ OROSCOPO
- ☐ COMPUTER MUSIC
- ☐ GESTIONE FAMILIARE
- ☐ BANCA DATI
- ☐ MATEMATICA FINANZIARIA/DISCO
- ☐ MATEMATICA FINANZIARIA/CASS.
- ☐ MATEMATICA FINANZIARIA/RIVISTE
- ☐ ANALISI DI BILANCIO/DISCO
- ☐ ANALISI DI BILANCIO/CASS.
- ☐ ANALISI DI BILANCIO/RIVISTE
- ☐ DICHIARAZIONE DEI REDDITI/DISCO
- ☐ DICHIARAZIONE DEI REDDITI/CASS.
- ☐ ARREDARE/DISCO
- ☐ ARREDARE/CASSETTA
- ☐ GRAPHIC EXPANDER/DISCO

Valore complessivo: Lire..

Su tale importo mi praticherete lo sconto del 10% in quanto abbonato a ☐ *Commodore Computer Club* ☐ *Personal Computer* ☐ *Computer* ☐ *VR Videoregistrare.* Pertanto vi invio la somma soltanto di lire..

☐ Desiderando ricevere le copie ordinate con la massima urgenza, accludo assegno bancario n.ro..
Banca.. per lire.. voi intestato.

☐ Contentandomi dei normali tempi postali ho inviato oggi stesso l'importo di lire........................ a mezzo C/C postale N. 37952207 intestato a Systems Editoriale - V.le Famagosta, 75 - 20142 Milano.

Ritagliare e spedire in busta chiusa regolarmente affrancata a Systems Editoriale - V.le Famagosta, 75 - 20142 Milano.

Nome ....
via .... **N.ro.** .... **telefono** ....
CAP .... **Città** ....

La Grande Libreria Systems

Autori Vari

64 Programmi per Commodore 64

Giochi, grafica, gestione delle stringhe, musica, numeri, gestionali.

Lire 4.800

Autori Vari

I miei amici C16 & Plus4

Un manuale pratico per padroneggiare il basic di questi computer.

Lire 7.000

Autori Vari

Strategie vincenti per Commodore 64

Le strategie per tutti i classici del videogioco: per giocarli, vincerli o programmarli.

Lire 5.800

Autori Vari

62 Programmi per il Vic 20, C16 e Plus 4

Giochi, grafica e routine per imparare a programmare.

6.500

Roberto Didoni, Guido Grassi

Utilities e giochi didattici

Raccolta di programmi pratici per tutti i Commodore e lo Spectrum.

Lire 6.500

Giovanni Mellina

Tutti i segreti dello Spectrum

4 passi nella Rom: come usare le più importanti routine del sistema operativo.

Lire 7.000

Roberto Didoni, Guido Grassi

Simulazioni e test per la didattica

Teoria e listati per Vic 20, C16, C64 C128 e Spectrum Sinclair.

Lire 7.000

Paolo Goglio

Impara giocando il basic dello Spectrum

Esercizi pratici per entrare nel vivo della programmazione.

Lire 7.000

Clizio Merli

µPascal per Commodore 64/128

Un manuale completo per il programma compilatore

Lire 7.000

Umberto Colapicchioni e Luca Galuzzi

Dal registratore al drive del C64

Tutti i segreti delle memorie di massa del Commodore 64

Lire 7.000

Autori Vari

ADA

Il linguaggio passepartout dei computer degli anni '80.

Lire 5.000

Clizio Merli

Il linguaggio PASCAL

Un manuale tascabile per lo studio e la programmazione.

Lire 5.000

Sì, voglio arricchire la mia biblioteca con i seguenti volumi al prezzo di copertina + lire 3.000 per spese di spedizione.

- ☐ **64 Programmi per Commodore 64**
- ☐ **Strategie vincenti per i tuoi videogames**
- ☐ **62 Programmi per Vic 20 C16 e Plus77**
- ☐ **Utilities e giochi didattici**
- ☐ **Tutti i segreti dello Spectrum**
- ☐ **Simulazioni e test per la didattica**
- ☐ **Imparare giocando il basic dello Spectrum**
- ☐ **I miei amici C16 e Plus4**
- ☐ **Pascal per Commodore 128**
- ☐ **Dal registrattore al drive del C64**
- ☐ **ADA**
- ☐ **Il linguaggio Pascal**

Nome ....

via .... **N.ro.** .... **telefono** ....

CAP .... **Città** ....

Su tale importo mi praticherete lo sconto del 10% in quanto abbonato a ☐ *Commodore Computer Club* ☐ *Personal Computer* ☐ *Com puter* ☐ *VR Videoregistrare.* Pertanto vi invio la somma soltanto di lire ....

Valore dell'ordine lire..............................

Ritagliare e spedire in busta chiusa regolarmente affrancata a Systems Editoriale - V.le Famagosta, 75 - 20142 Milano.

Entra nel grande Club

Fin dallo sbarco in Italia della Commodore ***Commodore Computer Club*** è il punto di riferimento di tutti gli utenti di C/64, Vic 20, C/16, Plus 4 ed ora di PC 10/20 ed Amiga.

Articoli didattici, recensioni e programmi istruttivi ed a basso costo hanno fatto di *Commodore Computer Club* la prima rivista italiana d'informatica.

Ma, per i lettori, Commodore Computer Club non è solo rivista: è consulenza telefonica gratuita, software originale pubblicato a latere dalla stessa casa editrice, un ponte verso l'informatica "maggiore" anche attraverso la collaborazione con le riviste sorelle *"Personal Computer"* e *"Computer"*.

E' per questa ragione che, anno dopo anno, aumenta il numero dei lettori che preferiscono ricevere la rivista in abbonamento invece di acquistarla in edicola. Ad essi l'editore riserva una serie di vantaggi esclusivi come:

- **un libro in omaggio** da scegliere tra i titoli disponibili della collana I libri di Systems*;
- **l'uso di una linea telefonica speciale** per richieste di consigli, e consulenza, il cui numero e le modalità d'uso verranno comunicate in forma riservata alla ricezione dell'abbonamento;
- **un canone annuo particolarmente interessante** di lire 40.000 per 11 fascicoli di Commodore Computer Club e di lire 35.000 per 11 fascicoli di Personal Computer;
- **l'esclusivo canone cumulativo** di lire 65.000 per 11 fascicoli di Commodore Computer Club ed 11 di Personal Computer;
- **uno sconto del 10%** su tutti gli acquisti per corrispondenza dei prodotti software su disco o cassetta, fascicoli arretrati o libri della Systems senza limiti di quantità.

** I titoli disponibili sono quelli reclamizzati sull'apposita pagina pubblicitaria "La libreria di Systems".*

Inviatemi in omaggio il volume della collana i libri di Systems..............................

Registrate oggi stesso il mio abbonamento a: ☐ Commodore Computer Club (Lire 40.000)
☐ Commodore Computer Club + Personal Computer (Lire 65.000)

☐ Desiderando ricevere le copie ordinate con la massima urgenza, accludo assegno bancario n.ro..............................
Banca.............................. per lire.............................. voi intestato.

☐ Contentandomi dei normali tempi postali ho inviato oggi stesso l'importo di lire .... a mezzo C/C postale N. 37952207 intestato a Systems Editoriale - V.le Famagosta, 75 - 20142 Milano.

Nome ....
via .... **N.ro.** .... **telefono** ....
CAP .... **Città** ....

Ritagliare e spedire in busta chiusa regolarmente affrancata a Systems Editoriale - V.le Famagosta, 75 - 20142 Milano.

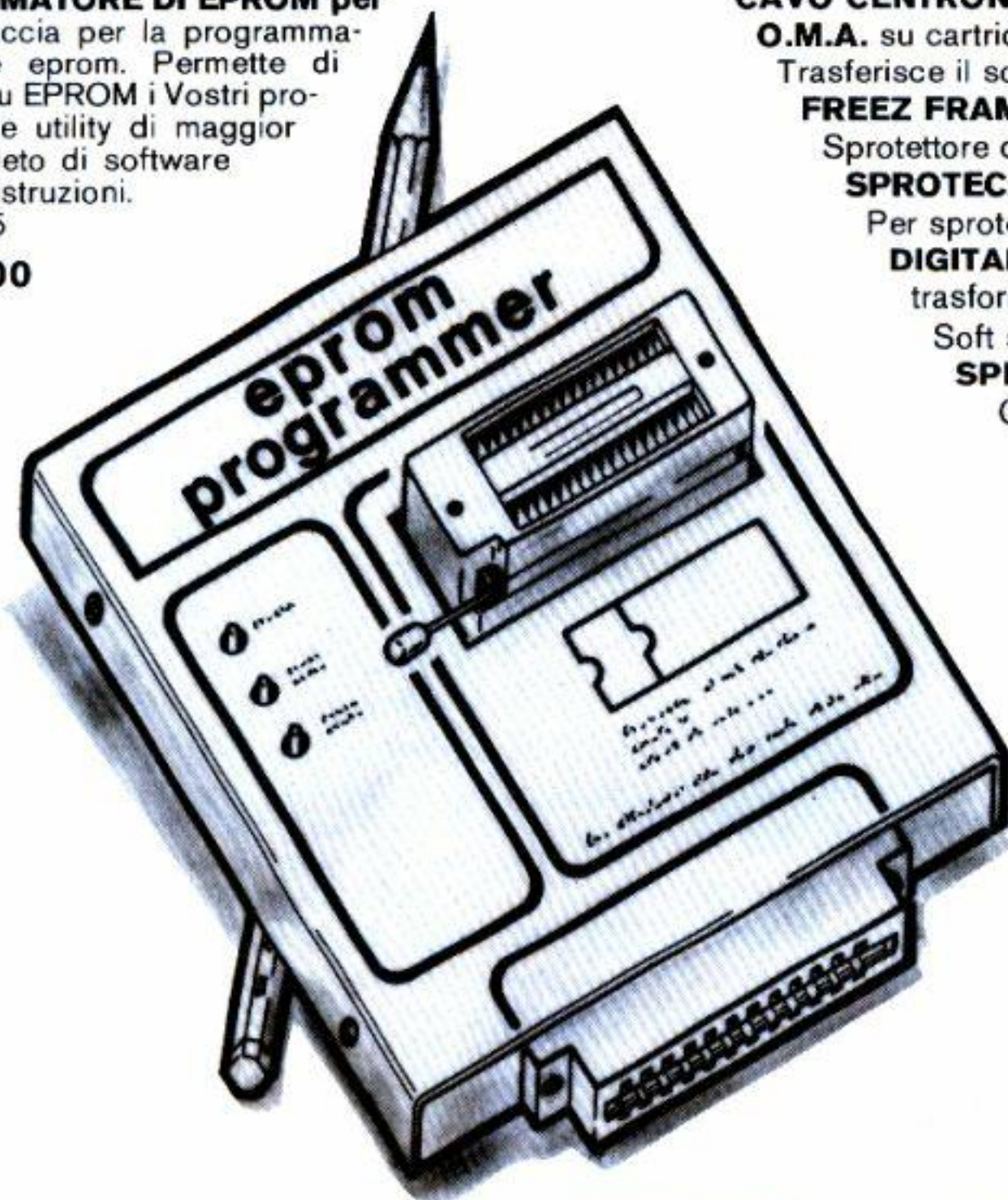
computer service

ACCESSORI PER COMPUTER HOME E PERSONAL COMPUTER

VENDITA PER CORRISPONDENZA

PROGRAMMATORE DI EPROM per c64. Interfaccia per la programmazione delle eprom. Permette di archiviare su EPROM i Vostri programmi o le utility di maggior uso. Completo di software su disco e istruzioni.
Art. CD 925
L. 160.000

studio ADVER

NOVITÀ

CAVO CENTRONICS AMIGA Art. CD 112	**L. 38.000**
O.M.A. su cartridge per c64 Art. CD 130 Trasferisce il soft protetto e TANTE altre utiliti.	**L. 60.000**
FREEZ FRAME per c64 Art. CD 132 Sprotettore di programmi su nastro e su disco	**L. 55.000**
SPROTEC/64 (isepic) Art. CD 910 Per sproteggere i programmi del c64	**L. 60.000**
DIGITALIZZATORE AUDIO per c64 Art. CD 915 trasforma le voci in segnali digitali. Soft su disco.	**L. 89.000**
SPEED CONTROLLER per c64 Art. CD 920 Cartuccia per ottenere l'effetto moviola.	**L. 35.000**
CARTRIDGE DI PROGRAMMAZIONE EPROM per il CD 925. Art. CD 930. Evita il caricamento del soft dal disco.	**L. 50.000**
MODEM 300 baud per c64 Art. CD 905	**L. 156.000**
MOUSE-LOGIMOUSE C7 - 3 TASTI con software per PCXT Art. PC 365	**L. 275.000**
MODEM V21 V23 seriale per PCXT Art. PC 375	**L. 360.000**

Articolo	Art.	Prezzo
Duplicatore cassette Copia con un registratore normale. Per c64 c128 vic20	Art. CD 102	**L. 30.000**
Copiatore programmi Copia con due registrari commodore. Per c64 vic20 c128	Art. CD 103	**L. 30.000**
Interfaccia radio Collega la radio al computer. Per c64 c128 o vic20	Art. CD 104	**L. 30.000**
Kit allineamento registratori c64 c128 vic20 Kit con strumento indicatore, nastro e cacciavite.	Art. CD 105	**L. 45.000**
Alimentatore per c64 e vic20	Art. CD 106	**L. 38.000**
Batteria tampone con batterie ricaricabili - Alimenta il c64 o vic20 in assenza di corrente per 30'	Art. CD 107	**L. 118.000**
Commutatore antenna tv/computer	Art. CD 108	**L. 9.500**
Tasto reset per c64 vic20	Art. CD 109	**L. 5.500**
Turbo Dos Velocizza il drive di circa 6 volte. Per commodore 64	Art. CD 115	**L. 35.000**
Penna ottica grafica per c64 (soft su disco)	Art. CD 121	**L. 39.000**
Penna ottica grafica per c64 (soft su nastro)	Art. CD 125	**L. 39.000**
Cuffia per Commodore per vic20 c16 c64 c128	Art. CD 150	**L. 19.000**
Copritastiera in plexiglass per c64 c16 vic20	Art. CD 750	**L. 13.000**
Copritastiera in stoffa per c64 c16 vic20	Art. CD 760	**L. 10.000**
Vaschetta floppy in plexiglass (x 40 pz. con chiave)	Art. CD 770	**L. 30.000**
Vaschetta floppy in plexiglass (x 90 pz. con chiave)	Art. CD 780	**L. 37.000**
Kit pulizia testine registratore	Art. CD 815	**L. 13.500**
Kit pulizia disk drive	Art. CD 820	**L. 20.000**
Kit pulizia video antistatico	Art. CD 825	**L. 12.000**
Kit pulizia tastiera	Art. CD 830	**L. 16.500**
Foratore disk in plastica	Art. CD 840	**L. 10.000**
Speed dos plus Kit Velocizza il floppy di circa 20 volte. Per c64	Art. CD 900	**L. 68.000**
Eprom 2764 utilizzabile con l'articolo CD 925	Art. CD 950	**L. 8.000**
Eprom 27128 utilizzabile con l'articolo CD 925	Art. CD 952	**L. 12.000**
Stabilizzatore elettronico di tensione 500 W Con filtri e protezioni.	Art. CD 160	**L. 430.000**
Adattatore joystick per c16	Art. CD 225	**L. 10.500**
Adattatore registratoreper c16	Art. CD 226	**L. 19.500**
Nastro inchiostrato per MT80	Art. CD 610	**L. 14.000**
Nastro inchiostrato per Tally MT180	Art. CD 611	**L. 16.500**
Nastro inchiostrato per Tally 1000 e Honeywell	Art. CD 612	**L. 9.500**
Nastro inchiostrato per Commodore MPS 801	Art. CD 614	**L. 13.000**
Nastro inchiostrato per Commodore MPS 802	Art. CD 616	**L. 15.000**
Nastro inchiostrato per Commodore MPS 803	Art. CD 618	**L. 18.000**
Pacco carta lettura facilitata 24"x11" 500 fogli	Art. CD 630	**L. 13.500**
Supporto stampante in plexiglass "fume" normale	Art. CD 660	**L. 45.000**
Supporto stampante in plexiglass "fume" rinforzato	Art. CD 670	**L. 57.000**
Disk 5" Singola Faccia Doppia Densità - 10 pezzi	Art. CD 700	**L. 25.000**
Disk 5" Doppia Faccia Doppia Densità - 10 pezzi	Art. CD 702	**L. 30.000**
Disk 3" Singola Faccia Doppia Densità - 10 pezzi	Art. CD 703	**L. 60.000**
Nastri magnetici C10 digitali 10 pezzi	Art. CD 712	**L. 20.000**
Nastri magnetici C15 digitali 10 pezzi	Art. CD 714	**L. 21.000**

SCONTI AI SIGNORI RIVENDITORI
TUTTI I PREZZI SONO COMPRENSIVI DI IVA. NON SI ACCETTANO ORDINI INFERIORI A L. 30.000.

BUONO DI ORDINAZIONE

04187

NOME

COGNOME

INDIRIZZO N.

C.A.P. CITTÀ

PROVINCIA P. IVA e/o Cod. Fisc.

VOGLIATE INVIARMI IN CONTRASSEGNO

Qt.	Cod. Art.	L.
Qt.	Cod. Art.	L.
Qt.	Cod. Art.	L.
Qt.	Art.	L.
PAGHERÒ AL POSTINO		L.

PIÙ SPESE POSTALI.

PER ORDINI TELEFONICI: 0522/661471-661647

computer service s.n.c.

Via B. Cellini, 4 - 42017 NOVELLARA (R.E.) tel. 0522/661647

C'è Monitor e monitor

□ Che cosa è il programma Monitor?
(Gianluca Cipriani - Collarmele)

• Per monitor (oltre all'apparecchio per visualizzare) si intende un programma di utilità che consente di lavorare in Linguaggio Macchina e/o in Assembly (=linguaggio... del linguaggio macchina).

Non esiste un solo programma Monitor, come non esiste un solo programma di Word Processor (=trattamento di testi). Possiamo dire, piuttosto, che Monitor è una "categoria" di programmi.

Alcuni di questi possono essere considerati standard; per esempio, sul C/16, sul Plus/4 e sul C/128 sono presenti, nelle stesse Rom, programmi di questa categoria, che sono richiamabili digitando, semplicemente...

Monitor

...e premendo il tasto return. Naturalmente, per usarli, è indispensabile conoscere il linguaggio macchina del microprocessore su cui girano, la mappa della memoria del computer e altre cose edificanti ed istruttive.

Stampa minga

□ La mia stampante (non originale Commodore, ma reclamizzata come 803 compatibile), crea problemi sia quando adopero Easy Script che quando uso un particolare programma di contabilità (allego output di stampante). Quale può essere la causa?
(Gianfranco De Lucia - Somewere)

• Molte stampanti MPS-803 compatibili posseggono, oltre alle peculiarità della 803, altre caratteristiche che devono essere "settate" opportunamente (via hardware oppure software) per un corretto funzionamento; alcuni codici particolari di Easy Script (stampa allargata, in neretto eccetera) è probabile che corrispondano, invece, ad un carattere alfabetico speciale: si spiegherebbe, dunque, la presenza del carattere U' invece della funzione.

Per ciò che riguarda il programma di contabilità, di cui alleghi la fattura di esempio, il motivo è dovuto quasi certamente al fatto che il programma stesso richiede una stampante in grado di stampare più di 80 colonne (tipicamente: 132 colonne). La tua stampante, che può riprodurre solo 80 colonne, è "costretta" ad andare a capo tutte le volte che si chiede di stampare un rigo di lunghezza maggiore. Lo stesso difetto, come puoi intuire, si manifesterebbe anche con una 803 originale Commodore.

Se, però, la stampante dispone di interruttori che modificano il numero di caratteri riproducibili, puoi settarla opportunamente ad una lunghezza maggiore; è ovvio che, in questo caso, la dimensione dei caratteri risulta minore e questo fatto potrebbe impedire l'uso di moduli prestampati che richiedono dimensioni diverse.

Chi non risica non rosica

□ La procedura di protezione descritta su CCC N.36 (che richiede l'estrazione del disco durante la formattazione) non rischia di danneggiare il drive oppure il disco stesso?
(Giampaolo Cavallaro - Messina)

• In effetti tutti i manuali sconsigliano di effettuare l'operazione descritta; per ciò che riguarda il disco c'è da dire che la protezione consiste proprio nel... rovinare il disco (ma in modo softawre, non hardware); per il drive, invece, l'unica cosa che può accadere è il momentaneo "accecamento" del sistema operativo del disco, che ricerca disperatamente una traccia magnetica e, per riuscire nell'intento, fa sbattere la testina contro il fine corsa. Quest'operazione, comunque, viene effettuata normalmente tutte le volte che si formatta un disco vergine: il drive è stato progettato (e costruito) per resistere a tali sollecitazioni meccaniche.

Per proteggere, comunque, si fa questo ed altro...

Mouse analogico

□ Con il programma dimostrativo pubblicato a corredo dell'articolo "Come costruire un mouse" (CCC N.34) non è possibile spostare lo sprite a destra oltre un certo punto dello schermo. Come fare per arrivare all'estrema destra?
(Alex Acquarone - Genova)

• Lo schermo è "largo" 80 caratteri; poichè ogni carattere è formato da una griglia di 8x8 puntini (=pixel), ne consegue che è largo 320 pixel (=40x8).

Le caratteristiche di uno sprite, tra cui la sua posizione, sono gestibili per mezzo di istruzioni Poke che, come argomento, accettano un valore massimo di 255. Per spostare gli sprite oltre il valore 255 (orizzontale), è quindi necessario utilizzare una seconda locazione (la 53264) i cui otto bit rappresentano le otto informazioni per i relativi sprite.

Se il bit di questa locazione vale zero, lo sprite corrispondente avrà come ascissa il valore dato dalla sua locazione "normale".

Se, invece, vale 1, lo sprite in oggetto sarà posizionato, orizzontalmente, a 256 pixel oltre quelli memorizzati nella corrispondente locazione "normale".

Il programma cui ti riferisci aveva solo scopo dimostrativo, e non ho ritenuto opportuno complicarlo aggiungendo la possibilità di andare oltre il 255simo pixel di destra.

Angoscia

□ Dopo aver copiato il listato sull'archivio Lotto (CCC N.37) mi sono accorta (e solo dopo aver memorizzato ben 900 numeri!) che i dati sono stati registrati "spostati" di 12. Se non mi aiuterete penso che l'unica alternativa è quella di dedicarmi alla maglia ai ferri...
(Nicoletta Leonardi - Avenza)

• Ma nooo! se non riesci a rintracciare l'errore (che hài commesso, chissà dove, nel trascrivere il listato) puoi sempre servirti del programma dal momento che sai che l'intervallo (fisso) è 12: basterà tener presente questa

anomalia per interpretare correttamente i dati elaborati.

Se proprio non ti rassegni, comunque, tieni presente che mi piacerebbe un bel pullover di color grigio perla; la mia taglia è la 46/48...

Risposte Rapide

Cuffietta

Per dotare di una presa per cuffia il monitor 1902 è necessario, come per qualsiasi monitor o TV, manomettere il circuito dell'altoparlante e collegare tra loro i due pin (lato cuffia) della presa stereo. Superflua la solita raccomandazione: se non te ne intendi di montaggi elettronici, lascia perdere!
(Marco Gigante - Cavallino)

Malignità

Il commerciante che sottolinea la presenza di difetti nel nuovo C/64 che, secondo lui, non sono presenti sul modello vecchio che ti vuole vendere, mente sapendo di mentire.
(Anonimo del '900)

Nessuna meraviglia

Non è esatto che mi meraviglio quando mi accorgo della giovanissima età di alcuni lettori (dodicenni in particolare); quando metto in evidenza l'età, anzi, lo faccio solo per dimostrare che l'informatica è alla portata di una fascia di età che sembra non conoscere limiti. Se poi, come affermi, la tua città pullula di dodicenni appassionati di computer, non posso che rallegrarmene.
(Andrea Micelli - Udine)

Scopo umanitario

Non mi è molto chiaro il programma di cui l'associazione che rappresenti ha bisogno, ma poichè lo scopo è dei più nobili, telefonami per verificare se posso aiutarti; naturalmente gratis.
(Claudio Romeo - Milano)

Quale Sys?

Non si possono impartire impunemente Sys casuali senza aspettarsi imbarazzanti conseguenze (suoni buffi, scomparsa video, comparsa Monitor, blocco totale del computer).
(Alessandro Orlando - Palermo)

Variabili

Per gestire correttamente le variabili è necessario uno studio approfondito, affrontato sui N.31, 32 e, in parte, 34 della nostra Rivista.
(Fausto Salvatore - Lamezia terme)

Quale sintassi?

Non specifichi con precisone quale è la sintassi che hai adoperato per effettuare il Backup e, di conseguenza, non posso esserti di alcun aiuto.
(Christian Capone - Torvajanica)

Swap?

L'istruzione Swap serve per scambiare tra loro il contenuto di due variabili. Se, ad esempio, "A" vale 10 e "B" vale 32, con l'istruzione Swap A,B i valori rispettivi vengono scambiati tra loro.
(Simone Malaspina - Genova)

Geos?

Forse tra breve la Commodore si deciderà a commercializzare liberamente il programma Geos, il manuale in italiano ed il mouse che aiuta nella gestione del software. Porta pazienza...
(Mario Barbaro - Bologna)

Schede deturpanti

Non è necessario strappare la scheda posta in fondo alla rivista per i quesiti di questa rubrica: è sufficiente una lettera purchè sia chiara e, soprattutto, breve!
(da alcune telefonate)

Compatibilità 1541 e 1541C

I due drive sono compatibili a livello software; la differenza consiste in una diversa disposizione dei componenti sul circuito stampato.
(Elvis Neri - Forlimpopoli)

Ben gentili!

Ringrazio per le gentili espressioni usate nei riguardi dei collaboratori di CCC.
(Sergio Bono - Cuneo; Edoardo Mingolla - Napoli)

Che memoria!

Per sapere di quanti byte disponi è sufficiente digitare Print Fre(0). Se il valore che viene visualizzato è negativo, devi digitare: Print 65535+ Fre(0).

Altre stampanti

Lascia perdere quello che ti dice il negoziante e accontentati delle funzioni di cui dispone la tua stampante non 803 compatibile. La colpa, infatti, non è della stampante, che per lavori di Word processor mi sembra buona, ma del negoziante, che non ti ha avvisato, prima di vendertela, degli inconvenienti che avresti incontrato usando la grafica in alta risoluzione.
(Mauro Strinna - Lepanto)

Aiuto impossibile

Come vedi, nello spazio della "Posta" c'è posto anche per te; per quanto riguarda il listato cui ti riferisci, purtroppo, non posso sostenere altra tesi se non un errore di trascrizione da parte tua.
(Bianco Ivano - Macerata)
(Stefano Scandella - Clusone)

Joystick

Non è possibile apportare semplici modifiche per fare in modo che un gioco, previsto per la sola tastiera,

possa esser gestito anche dal comodo accessorio.
(Giovanni Rossi - Imperia)

Trenini

Un plastico ferroviario può esser gestito tranquillamente tramite la User Port che dispone di otto bit paralleli (e non seriali, come sostieni) selezionabili a volontà in ingresso oppure in uscita. Se, poi, applichi anche un decodificatore, il numero di segnali gestibili risulta decisamente superiore.
(Mario Manotti - Roma)

Video oscillante

Se l'immagine del Tv ondeggia, è probabile che la causa sia da imputare a oscillazioni sul suo circuito di stabilità video. Prova a tener lontano dall'apparecchio qualsiasi fonte di disturbo (tra cui il trasformatore del computer); in caso contrario sarebbe bene provvedere all'acquisto di un monitor o di un moderno TV, dopo esserti accertato che la causa del difetto non risieda nello stesso computer.
(Roberto Vassallo - Melzo)

Meglio di no

Ti sconsiglio di collegare al C/64 (difettoso nella sua uscita TV) il modulatore del Vic 20.
(Nuccio Zarzana - Partanna)

Senza offesa

Nel numero scorso ("L'altra Posta") non volevo offendere nessuno nè, tantomeno, suggerire "interventi" di tipo afgano sul suolo patrio; le riflessioni in merito ai risultati offerti dal nostro sistema scolastico, infatti, scaturiscono da considerazioni fondate su insoddisfazioni generali, riscontrabili leggendo un qualsiasi periodico.

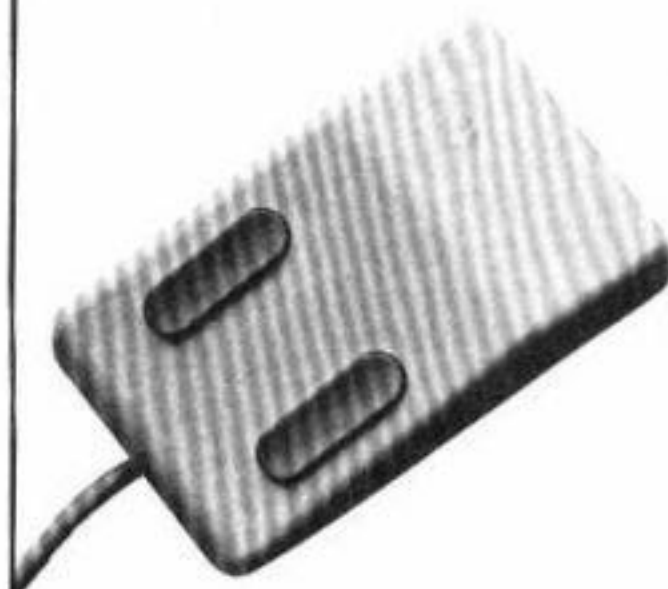
Se, poi, l'ironia che affiora in più parti ha dato fastidio a qualcuno, sarebbe opportuno considerare il sarcasmo delle categorie interessate (industria e commercio, vale a dire coloro che offrono i posti di lavoro) sugli stessi argomenti.

Toma

Le routine grafiche di Toma girano esclusivamente sul C/64, e non sul C/16.
(Stefano Morgagni - Forlimpopoli)

a sole 285.000 Lire, IVA compresa
il Disk Drive per il tuo COMMODORE 64/128

1) Compatibile al 100%, 2) Slim Line, 3) Facile deviatore esterno per cambiare il Numero del Drive, 4) Robusto mobile in metallo, 5) Due connettori seriali, 6) Garanzia totale.

MOUSE per C 64/128
a un prezzo INCREDIBILE !

90.000 Lire

GRATIS con il MOUSE il programma su disco per utilizzarlo al meglio.

IN OMAGGIO un dischetto con i migliori programmi TURBO per trasferire su disco tutti i giochi e utilities che hai su cassetta !!!

Spedizioni in tutta Italia con pagamento contrassegno al postino + L. 15.000 per spese di spedizione. Nessun addebito di spese a chi allega all' ordine un assegno non trasferibile intestato alla CIRCE Srl - Gratis inviamo il Ns. Catalogo HARDWARE.
CIRCE Srl - Via 1° Maggio, 26 - Zona Industriale 37012 Bussolengo ,(VR) -
Tel. 045/71.51.043

Software made in Italy

Una nuova iniziativa della Systems Editoriale che consentirà ai nostri lettori più in gamba di guadagnare qualche milioncino...

di Alessandro de Simone

Diciamo la verità: siamo stufi di veder girare software, pirateggiato o meno, che proviene quasi esclusivamente dagli Stati Uniti.

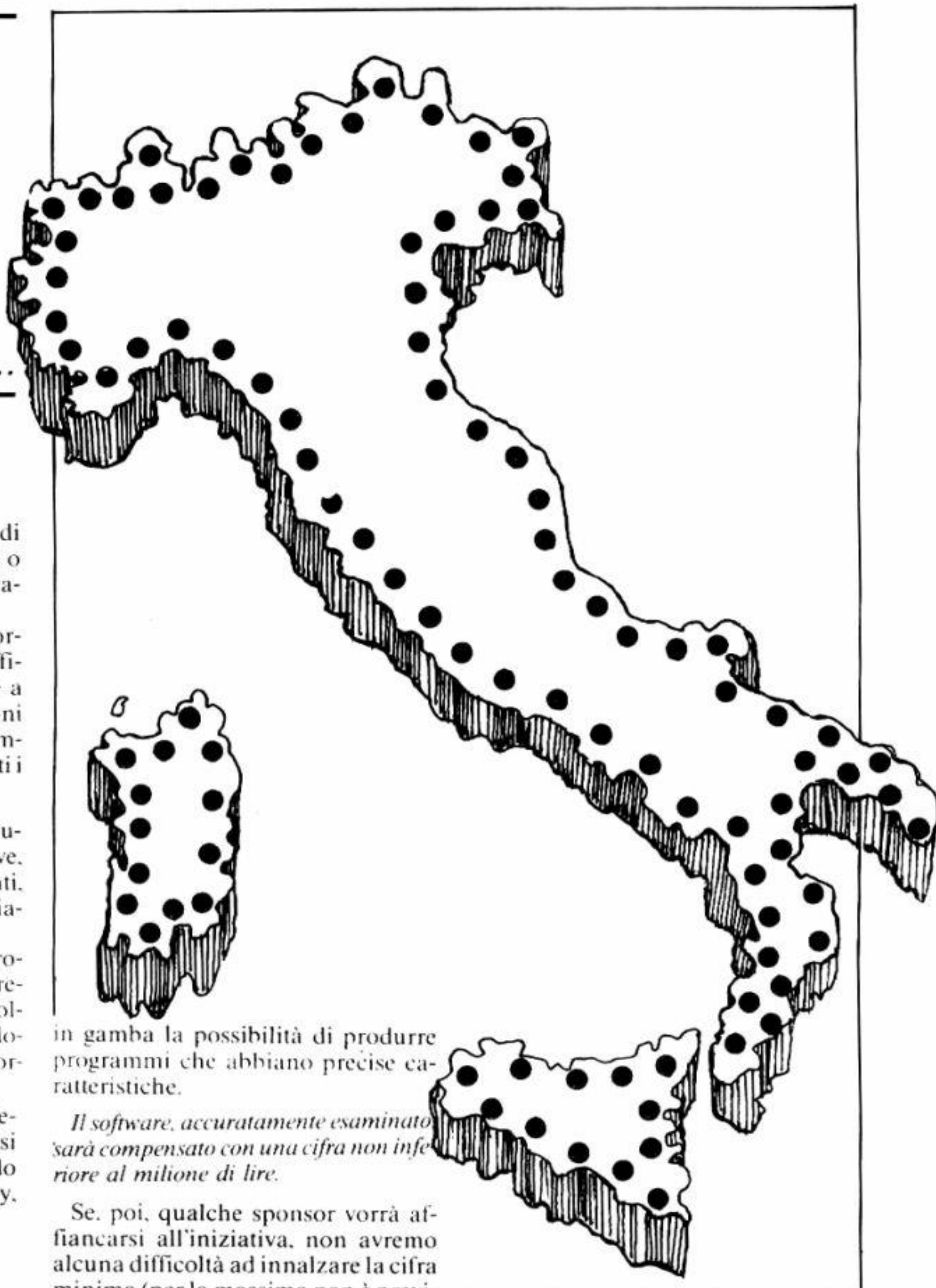
In Italia, e non c'è bisogno di ricordarlo, abbiamo idee e fantasia sufficienti per far mangiare la polvere a chiunque. Continue dimostrazioni sono offerte dall'assegnazione di ambiti premi per studi e ricerche in tutti i campi dell'umano sapere.

Il software, ed il campo dei computer in generale, non può, e non deve, restare al di fuori di riconoscimenti, più o meno espliciti, ai quali possiamo sicuramente aspirare.

E' ben vero che gli americani producono programmi di notevole pregio, ma è anche vero che noi, una volta entrati in possesso della giusta documentazione, siamo in grado di fornire software più che dignitoso.

Come è noto, e scusate l'immodestia, la Systems editoriale da tempo si pone in prima linea proponendo software, interamente made in Italy, di tutto rispetto.

La proposta

L'idea nata in Redazione è piuttosto semplice: dare a tutti i lettori più in gamba la possibilità di produrre programmi che abbiano precise caratteristiche.

Il software, accuratamente esaminato sarà compensato con una cifra non inferiore al milione di lire.

Se, poi, qualche sponsor vorrà affiancarsi all'iniziativa, non avremo alcuna difficoltà ad innalzare la cifra minima (per la massima non è previsto un limite...) o a fornire gratuitamente apparecchi e servizi.

Ma è giunto il momento di parlare concretamente.

Il non-concorso

Dovrebbe esser chiaro che questo non è un bando di concorso: la fortuna non c'entra per nulla, nè sono previsti posti di lavoro da coprire in base all'esito di faraonici concorsi da svolgersi all'interno di stadi e palazzetti dello Sport.

La possibilità di darsi da fare è offerta a tutti indistintamente; anzi, per evitare che il tempo sia un fattore determinante, non accetteremo i lavori prima di una certa data.

Ci farebbe piacere festeggiare il N.50 della nostra rivista, Commodore Computer Club, con la pubblicazione dei nomi degli autori del software.

Il tempo c'è, e tanto; l'importante è buttarsi a capofitto nella nostra iniziativa.

Le caratteristiche del software

Il software deve avere caratteristiche ben precise che tengano conto, in ogni caso, delle seguenti indicazioni:

• I programmi possono essere di qualunque tipo (utility, word processor, data base, spreadsheet, giochi di alto livello, simulazioni, intelligenza artificiale, linguaggi, compilatori eccetera) senza limiti di applicazioni.

• Possono essere scritti in qualsiasi linguaggio purchè sia possibile caricarli in macchina senza esser costretti a caricare il linguaggio stesso, a meno che la Systems Editoriale non possegga i diritti di copyright sul linguaggio.

Un programma scritto con il Basic ufficiale Commodore (di cui sono dotate tutte le macchine) oppure con il nostro Gw-Basic oppure con le routine di Toma, ad esempio, verrà accettato; se scritto con il Simon's Basic oppure con il Logo non potrà essere preso in considerazione.

• Le macchine su cui gira il software devono appartenere ad una delle tre categorie: C/64 e C/128; Amiga; Calcolatori Commodore della serie PC.

Sono quindi esclusi i vecchi elaboratori della serie Pet, il Vic-20, il C/16, il Plus/4 e, ovviamente, tutti gli altri calcolatori di marca non Commodore.

• Il software deve tassativamente essere totalmente "aperto": chiunque deve avere la possibilità di modificarlo senza ricorrere ad alterazioni eccessive.

ne consegue che non deve essere prevista alcuna forma di protezione.

• Il software deve essere, nella versione definitiva, abbondandemente documentato per consentire sofisticazioni e migliorie da parte dei programmatori "evoluti".

• Pur se la Systems Editoriale vanterà i diritti commerciali sul lavoro consegnato, il software apparterrà alla categoria "Free": saranno possibili scambi tra utenti, ma *sarà considerata illecita qualsiasi forma di commercializzazione del prodotto che rimane di esclusiva proprietà della Systems Editoriale.*

• Chiunque entri in possesso del software in oggetto si deve impegnare a lasciare immutata la schermata di presentazione in cui comparirà la scritta "Software Made in Italy", il nome degli autori e quello di coloro che hanno apportato eventuali modifiche al programma originale.

• La presente opportunità è riservata *esclusivamente* ai cittadini italiani o di origine italiana.

• Per garantire eguali possibilità a chiunque intenda rispondere al presente annuncio, nessun lavoro sarà esaminato prima del prossimo novembre 1987.

• Per prendere accordi sul software da sviluppare, e per evitare di mettersi al lavoro su programmi ritenuti non interessanti, è indispensabile telefonare in Redazione (tel.02/ 84.67.34.8).

• Il giudizio, insindacabile, sulla qualità dei programmi presentati, sarà espresso da un apposito Team formato da nostri incaricati.

Systems Editoriale: sempre un passo avanti

TANTI BUONI MOTIVI PER ABBONARSI A

12 NUMERI AL PREZZO DI 10
solo 45.000 lire
invece di 54.000 lire

•

PREZZO BLOCCATO
per tutta la durata dell'abbonamento

•

SICUREZZA
di non perdere neanche un momento

•

COMODITÀ
di ricevere la propria rivista preferita a casa

•

COSA STATE ASPETTANDO?

Play e i suoi fratelli

Un gruppo di esperimenti utili per far suonare correttamente il C/128

di Alessandro de Simone

SCHEDA TECNICA

Note tecniche di tipo didattico per applicazioni musicali.

Articolo idoneo per il solo computer C/128 (in modo 128).

Consigliato ai principianti

Tempo fa, ed esattamente sul N.40 della nostra rivista, avevamo spiegato in dettaglio il modo di operare del C/128 con il comando più semplice che potesse riprodurre un suono: Sound e i suoi otto parametri.

Stavolta ci occuperemo dell'altro comando disponibile, Play, cercando di rendere il discorso semplice e completo, in modo da ampliare le (scarse) notizie finora diffuse dalla stessa Commodore e da altre pubblicazioni.

Sembra, infatti, che una notevole confusione circondi il potente comando che, a volte, non risponde come ci si aspetta. Inizieremo, pertanto, a trattare in modo approfondito l'argomento intrattenendoci brevemente anche su concetti che dovrebbero esser già noti ai lettori.

Le caratteristiche di un suono

Una qualsiasi emissione sonora possiede, tra le altre, alcune caratteristiche che la contraddistinguono dalle altre:

• *Volume*. E' l'intensità sonora (maggior volume, maggior "potenza").

• *Forma d'onda*. E' il "disegno" che l'emissione sonora traccerebbe sullo schermo di un oscilloscopio. Esiste, come è intuitivo, una varietà infinita di forme d'onda perchè è possibile sommare tra loro più forme d'onda ed avere un effetto totalmente diverso da quello generato dalle singole onde. Con il C/128 è possibile generare alcuni tipi fondamentali di forme d'onda: quella sinusoidale, quella triangolare, quella quadra e, infine, il "rumore".

• *Tempo*. Rappresenta il periodo di tempo "modulare" che viene assunto come base per la riproduzione successiva di note. Con il termine "tempo", quindi, non intenderemo ciò che comunemente si intende (tempo di rock, tempo di valzer e così via), ma unicamente il ritmo, o meglio la velocità di riproduzione di un brano. Potremo quindi riprodurre lo stesso brano a velocità differenti, limitandoci a variarne il tempo. E' ovvio che, ad esempio, uno slow suonato a velocità elevata perderà le sue caratteristiche principali (dolcezza, "romanticismo"); analogamente un motivo rock, suonato a velocità insufficiente, perde la caratteristica della ballabilità.

Strumento	*Attack*	*Decay*	*Sustain*	*Release*
Tamburo	*nullo*	*nullo*	*nullo*	*minimo*
Piatti	*nullo*	*nullo*	*nullo*	*elevato*
Chitarra	*nullo*	*nullo*	*nullo*	*elevato*
Violino	*variab.*	*variab.*	*variab.*	*variab.*
Pianoforte	*nullo*	*medio*	*nullo*	*variab.*
Organo	*nullo*	*nullo*	*massimo*	*variab.*
Flauto	*variab.*	*medio*	*variab.*	*variab.*

Il tempo, variabile tra zero (tempo lunghissimo) e 255 (brevissimo istante) è impostabile direttamente, come noteremo anche dagli esempi di queste pagine.

• *Attack.* E' il tempo che occorre affinchè l'intensità sonora dell'emissione sonora raggiunga il massimo. Un colpo di tamburo, ad esempio, ha un tempo di attack nullo in quanto si raggiunge subito la massima intensità. Un violino, a seconda di come viene suonato, può avere un tempo di attack nullo (corda "pizzicata", come la chitarra) oppure molto lungo, nel caso in cui il violinista, imprimendo all'arco una forza crescente, generi un aumento di volume suonando una stessa nota.

• *Decay.* E' il tempo che occorre affinchè l'intensità sonora, una volta raggiunto il volume massimo, diminuisca ad un livello più basso, per restare poi costante per l'intero periodo del sustain.

• *Sustain.* E' il periodo di tempo durante il quale l'intensità sonora, stabilizzatasi al livello del decay, rimane costante.

• *Release.* E' il tempo che occorre affinchè l'intensità sonora, partendo dal livello del sustain, si annulli.

I quattro parametri (detti, insieme, ADSR) hanno valori compresi tra 0 e 15.

Con il termine "variabile" si intende dire che il tempo può cambiare a seconda del musicista; suonando un flauto, ad esempio, è possibile realizzare un attack, un decay e un sustain nulli, o elevati, a seconda di come si soffia. Il sustain, invece, dipende dai polmoni...

Analogamente i quattro parametri, nel caso del violino, possono variare a piacimento, a patto di tener conto della lunghezza dell'archetto!

Un primo approccio

I comandi di "contorno" al comando Play sono numerosi e ci soffermeremo su quelli indispensabili per una sua corretta programmazione.

Vol, variabile tra 0 e 15, è il comando che setta (=impone) il volume e che permette di intervenire facilmente sull'intensità generabile. E' possibile regolare il volume sia direttamente, con il comando Vol, che inserendo un codice all'interno della stringa da suonare.

Envelope e Play, invece, sono i comandi che richiedono una trattazione piuttosto approfondita.

Le voci del C/128

Il C/128, per chi ancora non lo sapesse, è perfettamente identico al C/64 per ciò che riguarda il chip sonoro. In pratica non esiste alcuna differenza tra i suoni riproducibili da un C/64 e quelli generabili dal C/128 in modo 128, proprio perchè i circuiti elettronici sono perfettamente identici.

L'unica differenza consiste nel fatto che mentre, con il C/64, è necessario ricorrere a comandi del tipo Poke (oppure a Tool appropriati), con il C/128 è possibile ricorrere ad appositi comandi Basic.

Abbandonando il Basic, insomma, e programmando un suono in linguaggio macchina, un programma per C/64 è sostanzialmente identico a quello che gira sul C/128 e sono anche eguali, ovviamente, gli effetti riproducibili.

Indubbiamente, soprattutto per un principiante, è molto più comodo disporre di comandi Basic, ma questi, mal utilizzati, potrebbero creare problemi.

Il C/128, o meglio il suo chip sonoro, dispone di tre voci indipendenti, vale a dire che è possibile generare, nello stesso istante, tre suoni diversi.

I primi esperimenti

Accendete il C/128 (ovviamente in modo 128) e digitate:

Play "ABC"

Dovreste udire tre note, emesse in successione; in caso contarario regolate il volume del vostro televisore e insistete finchè non le udite.

Con la semplice prova descritta ci accorgiamo che non appena si accende il computer (ma anche dopo aver premuto Run/Stop e Restore) il volume è posto a 9, il tempo è settato automaticamente al valore 8, la voce attivata è la n.1, l'ottava è la n.4; degli altri due parametri (filtro e envelope, settati a zero), parleremo in seguito.

Suonare note musicali, dunque, è un'operazione analoga al comando Print in cui la stringa interessata, stavolta, rappresenta non un gruppo di caratteri da stampare, ma note da suonare, secondo la scala musicale inglese che, come è noto, utilizza le lettere dell'alfabeto al posto dei nostrani Do, Re, Mi eccetera.

Oltre alle note musicali è possibile inserire altri codici che il computer interpreta seguendo un'opportuna simbologia:

W (whole: intero)
H (half: metà)
Q (quarter: un quarto)
I (eighth: un ottavo)
S (sixteenth: un sedicesimo)

Provate, in successione, la differenza tra i comandi che seguono, e prestate la massima attenzione alla durata di ciascuna nota tenendo presente, anche, che eventuali spazi presenti all'interno della stringa da suonare vengono semplicemente ignorati:

Play "ABC"

Play "IA BC"

Play "SA HB WC"

Play "ABC"

Con il primo comando le tre note vengono emesse secondo i parametri di default (cioè standard). Il secondo comando, invece, grazie alla presenza del codice "I", ripete la stessa sequenza, ma ad una velocità maggiore (la durata di ciascuna nota, infatti, è ridotta ad un ottavo).

Il terzo comando imposta tre tempi diversi per ciascuna nota: ad "A" è associato "S", per "B" vale "H" e per "C", invece, "W". L'ultimo comando dovrebbe riprodurre gli stessi effetti del primo ma... sorpresa! Possiamo quindi concludere che:

• La durata di una qualsiasi nota coincide con l'ultimo tempo impostato.
• Premendo Run/Stop e Restore si annulla qualsiasi impostazione precedentemente settata e vengono riattivati automaticamente i parametri di default.
• La durata di una nota dipende sia dal parametro "Tempo" che dal codice associato a ciascuna nota (W, H, Q, I, S); in pratica è possibile avere lo stesso effetto impostando valori brevi del ritmo (es.: Tempo 200) e note "lunghe" (es: Play"WA") o viceversa.

Ora premete Run/Stop e Restore e digitate i seguenti comandi:

Play"WA HBC"

Tempo 40

Play"WAHBC"

La nota "A" è suonata per un tempo intero, mentre "B" e "C" per metà. Il comando "Tempo" (settabile tra 0 e 255) non fa altro che stabilire, come abbiamo detto, una "base" che viene presa come riferimento per l'emissione sonora; in altre parole, qualunque sia il valore attribuito a "Tempo", il rapporto reciproco tra la durata delle note sarà sempre lo stesso.

Anche Tempo, come tutti gli altri parametri, viene riportato al valore di default (8) premendo Run/Stop e Restore. Un eventuale comando Run, come vedremo, non altera i parametri impostati.

La contemporaneità

Premete Run/Stop e Restore e digitate la seguente riga:

Tempo 8: Play"WA": Print"prova": Play"B"

Non appena premete il tasto Return, noterete l'emissione della nota intera "A" e, immediatamente dopo, la visualizzazione della parola "Prova"; dopo alcuni istanti sentirete la seconda nota "B", evento accompagnato dalla riapparizione immediata del cursore lampeggiante.

Ciò dimostra che, non appena un comando sonoro viene riconosciuto e attivato dal Basic, il sistema passa ad interpretare l'eventuale comando successivo (Print "prova", nel nostro caso). Il Basic passerebbe volentieri, a questo punto, ad interpretare il successivo comando (Play"B"), ma incontra, appunto, un nuovo Play riferito alla stassa voce di prima (non dimentichiamo che, salvo esplicite dichiarazioni contrarie, la voce di default è sempre la N.1). Siccome il comando precedente conteneva il codice "W" (nota intera), l'interprete Basic attende che la nota "A" venga emessa per intero, prima di emetterne un'altra.

Ciò accade, però, se pretendiamo di riprodurre un'altra nota con la stessa voce e se il tempo di emissione della nota precedente non è trascorso per intero.Premete Run/Stop e Restore e digitate quanto segue:

Tempo 4: Play"WV1A V2B V3C": Print"prova"

Notiamo la presenza di tre nuovi codici (V1, V2, V3) che, l'avrete intuito, servono per indicare al Basic la voce che sarà interessata dalla successiva nota musicale; ciò significa, quindi, che il comando Play riprodurrà, per intero (W), la nota "A" con la voce 1, la "B" con la 2 e la "C" con la n.3.

A differenza del caso precedente, infatti, il Basic non attende che la nota "A" venga emessa per intero prima di riprodurre la "B" e la "C" perchè queste note si riferiscono ad altre voci che, in quel momento, sono "libere".

Se, infatti, siete lesti di mano e, non appena premete il tasto Return, risalite rapidamente con il cursore sul rigo di comando e premete nuovamente Return, vi accorgete che il computer rimane muto finchè il tempo impostato (combinazione di Tempo 8 e "W") non è trascorso per intero.

In questo modo, quindi, è possibile suonare accordi, pur se con la limitazione di tre note per volta. E' ovvio che, dal momento che il Basic procede linearmente, istruzione dopo istruzione, le note non vengono emesse contemporaneamente, ma una alla volta, a mano a mano, cioè, che vengono interpretate dal Basic. Nonostante ciò possiamo affermare che, nella maggior parte dei casi, le note sembrano essere riprodotte contemporaneamente. Lavorando in Linguaggio Macchina la differenza si riduce ad alcuni millesimi di secondo e risulta inavvertibile.

Envelope Number	Instrument	Attack	Decay	Sustain	Release	Wave-form	Width
0	Piano	0	9	0	0	2	1536
1	Accordion	12	0	12	0	1	
2	Calliope	0	0	25	0	0	
3	Drum	0	5	5	0	3	
4	Flute	9	4	4	0	0	
5	Guitar	0	9	2	1	1	
6	Harpsichord	0	9	0	0	2	512
7	Organ	0	9	9	0	2	2048
8	Trumpet	8	9	4	1	2	512
9	Xylophone	0	9	0	0	0	

Ancora sui tempi

Resettate il C/128 e digitate più di una volta il seguente comando allo scopo di riconoscere, in seguito, la nota emessa:
Play "SA"

Ora scrivete il seguente programma:

100 Tempo 8
110 For I=1 to 4
120 Play"V1 SA V2 WB V3 WC"
130 Next

Ci aspettiamo di ascoltare, per quattro volte (For...Next), l'accordo "A B C" riprodotto dalle tre voci. Ciò che ascoltiamo, invece, è un primo accordo, per intero, e subito dopo la nota "A", isolata; il secondo accordo, se prestate attenzione, è formato PRIMA dalle due note "B" e "C" e DOPO dalla nota "A" (si tratta, comunque, di decimi di secondo di differenza). Ciò avviene perchè la voce 1, alla quale è affidato il compito di eseguire "A", risulta "libera" dopo appena un sedicesimo di battuta (notare "SA"); ne consegue che il computer, dopo aver riprodotto "B" e "C", passa ad eseguire l'istruzione Next. Siccome la voce 1 è libera, subito provvede a riprodurre la nota ad essa associata; le voci 2 e 3, al contrario, sono ancora impegnate ad eseguire il ciclo "intero" (WB WC) e costringono il computer ad attendere.

Se, però, modificate la riga 120 come segue...

120 Play"V1 SA V2 WB V3 WC M"

...il "difetto" di prima scompare perchè il codice "M" obbliga il Basic ad una pausa fintantochè il tempo relativo alla battuta impostata non sia interamente trascorso. Tale accorgimento, insomma, impedisce che alcune note vengano emesse prima del necessario e si comporta come una sorta di direttore d'orchestra.

Prima di procedere ricordiamo che alcuni caratteri consentono di alterare la nota di un semitono.

Esempi:

Play "#A" (diesis)
Play "$A" (bemolle)
Play ".A" (durata +50%)
Play "R" (pausa)

La corrispondenza delle note di "base" è la seguente:

Do (C), Re (D), Mi (E), Fa (F), Sol (G), La (A), Si (B).

Per cambiare ottava il codice è "O"; il valore di default è 4. Esempio:

Play "O3 A O4 A O5 A"

Sentirete il "La" (A) della scala 3, 4 e 5 rispettivamente.

Il comando Envelope

Come abbiamo già detto, regolando opportunamente i valori di Attack, Decay, Sustain e Release, è possibile simulare un qualsiasi strumento musicale.

La sintassi completa relativa a Envelope è la seguente:

Envelope En, At, De, Su, Re, Wf, Pw

in cui En rappresenta il numero di inviluppo selezionato (da 0 a 9); At, De, Su, Re sono i parametri ADRS variabili, ciascuno, da 0 a 15; Wf è la forma d'onda desiderata (0= triangolare; 1= sinusoidale; 2= quadra; 3= rumore; 4= modulazione ad anello); Pw rappresenta, in pratica, la simmetria dell'onda quadra (valori da 0 a 4096) ed è attiva, ovviamente, solo se viene selezionata la forma d'onda quadra.

Nel C/128, per venire incontro ai principianti, sono disponibili 10 registri (numerati da 0 a 9) che corrispondono agli inviluppi relativi ad altrettanti strumenti musicali:

0: pianoforte
1: fisarmonica
2: calliope
3: tamburo
4: flauto
5: chitarra
6: clavicembalo
7: organo
8: tromba
9: xilofono

Ad ognuno di questi, come si nota dalla tabella, è associato il gruppo di parametri At, De, Su, Re, Wa (=waveform, forma d'inda) e Wi (=Width, ampiezza, attiva solo quando è settata la forma d'onda n.2).

In teoria, stando a quanto è riportato sul manuale, sarabbe sufficiente impostare il numero di Envelope per ottenere diverse riproduzioni sonore. In pratica ciò non avviene e ce ne possiamo render conto se tentiamo di riprodurre le stesse note con effetto pianoforte (envelope 0) e organo (envelope 7):

Envelope 0: Play "ABC"

Envelope 7: Play "ABC"

La riproduzione sonora risulta identica nei due casi; se, però, inseriamo nella stringa da suonare il codice relativo all'envelope ("T"), si ottiene ciò che si desidera:

Play "T0 ABC"

Play "T7 ABC"

Naturalmente è possibile realizzare miscugli fantasiosi:

Play "T0 ABC T1 ABC T3 ABC"

Ricordiamo che eventuali spazi presenti all'interno della stringa vengono semplicemente ignorati.

Allo stesso modo è possibile inserire, in una stessa stringa, gli altri codici che consentono varie funzioni, e che qui riassumiamo affiancando i valori minimo, massimo e di default:

V: voce (1-3) 1
O: ottava (0-6) 4
T: envelope (0-9) 0
U: volume (0-15) 9
X: filtro (0-1) 0

Il manuale suggerisce, nel caso di impostazione contemporanea di più codici, di inserirli seguendo l'ordine indicato nella tabella precedente.

Rimandando ad un prossimo articolo lo studio del filtro, concludiamo con alcune considerazioni:

- Il comando Envelope privo di parametri, benchè accettato senza segnalazione di errore, non produce l'effet-

to desiderato; è indispensabile ricorrere al codice da inserire nella stringa.

• Il comando Envelope agisce sulle tre voci e rimane lo stesso finchè non lo si cambia esplicitamente. Per fare in modo di ottenere un effetto diverso per ciascuna voce, è necessario cambiare Envelope prima di indicare la voce. Esempio:

100 Play "V1 T1 A V2 T2 B V3 T3 C"

Con il microprogramma di prima sentirete fisarmonica (T1), calliope (T2) e tamburo (T3) che suoneranno, rispettivamente, le note "A", "B", e "C" mediante le voci 1, 2 e 3.

• Modificando i parametri di Envelope relativi ad una forma d'onda predeterminata (vedi programma di queste pagine) non è più possibile ritornare ai valori di default se non con Run/stop e Restore. In pratica, all'accensione del computer, ad alcune locazioni di memoria Ram vengono assegnati i parametri indicati sul libretto di istruzioni. Un'eventuale alterazione del loro contenuto rimane costante nel tempo.

Il programma

Il listato di queste pagine, di notevole... banalità, indica semplicemente un metodo per studiare la variazione dell'emissine sonora al variare delle note, dei parametri ADSR, dell'ottava e così via.

Invitiamo il lettore a studiarlo per proprio conto e, in seguito, a scrivere motivetti musicali che, sfruttando le caratteristiche del comando Play, siano in grado di far apprezzare le notevoli potenzialità sonore del C/128.

```
100 REM I COMANDI ENVELOPE, VOL, PLAY, TEMPO
110 REM NEL COMMODORE 128 IN MODO 128
120 :
130 REM BY ALESSANDRO DE SIMONE
140 :
150 PRINT CHR$(147)
160 INPUT"VOCE     (1/3)";VC:IFVC<1ORVC>3 THEN160
170 INPUT"VOLUME (0/15)";VL:IFVL<0ORVL>15THEN160
180 INPUT"OTTAVA (0/6)";OT:IFOT<0 OR OT>6 THEN 160
190 INPUT"ENVEL. (0/9)";EN:IF EN<0 OR EN>9 THEN 160
200 INPUT" MODIFICHI (S/N)";MD$
210 IF MD$="N"THENPRINT:PRINT:PRINT:PRINT:PRINT:PRINT:GOTO280
220 INPUT"      ATTACK (0/15)";AT:IF AT>15THEN 160
230 INPUT"      DECAY  (0/15)";DC:IF DC>15THEN 160
240 INPUT"      SUSTAIN(0/15)";SU:IF SU>15THEN 160
250 INPUT"      RELEASE(0/15)";RL:IF RL>15THEN 160
260 INPUT"      WAVEFOR(0/3)";WF:IF WF>3 THEN 160
270 INPUT"      WIDTH (0/2048)";WI:IF WI>2048 THEN 160
280 INPUT"NOTE DA SUONARE";NT$:IFNT$=""THENNT$="CDEFGAB"
290 INPUT"TEMPO  (1/255)";TP:IF TP<1 OR TP>255 THEN 160
300 TEMPO TP
310 PLAY"V"+STR$(VC):REM SETTA VOCE
320 PLAY"O"+STR$(OT):REM SETTA OTTAVA
330 IFMD$="N"THEN350:REM ESCLUDE MODIFICHE
340 ENVELOPE EN,AT,DC,SU,RL,WF,WI
350 PRINT:PRINT:PRINT" EN AT DC SU RL WF WI":PRINT
360 PRINT"                                        "
370 PRINT EN;AT;DC;SU;RL;WF;WI
380 PLAY"T"+STR$(EN):REM SETTA ENVELOPE
390 PLAY"U"+STR$(VL):REM SETTA VOLUME
400 PLAY NT$:GOTO160:REM SUONA NOTA E RICOMINCIA
410 END

READY.
```

SUL SENTIERO DELLE GIUBBE ROSSE

Una vera esperienza di vita per i ragazzi/e oltre i 10 anni
Abbinare lo studio della Lingua Inglese, al contatto di una natura incontaminata
Una vacanza-studio unica ed indimenticabile, in uno scenario che non ha confronti.

Questo tipo di vacanza è indirizzato sia ai principianti, sia a coloro che hanno già maturato una conoscenza della Lingua Inglese, ma il denominatore comune è il reale contatto con la natura.

- Un viaggio di 19 giorni attraverso la Provincia dell'Ontario a bordo di un "Super Van" da 15 posti, con aria condizionata e stereo system, con l'assistenza di personale qualificato.

Ogni 10 partecipanti ci sono 4 persone di assistenza.
Le attività standard includono:

- partecipazione alla vita di campeggio, canoa, tracking, pesca, white-water rafting, ginnastica, nuoto e studio della Lingua Inglese.

Quest'ultimo aspetto sarà curato da insegnanti specializzati dello SHENKER INSTITUTE OF ENGLISH, con due ore di lezione al giorno, seguendo il METODO SANDWICH di GEORGE SHENKER.

- Viaggio Andata/Ritorno MILANO/TORONTO in classe turistica.
- Tre pasti al giorno dalla 1ª colazione del secondo giorno, al lunch del 17 giorno.*
- Full-Day Immersion di Lingua Inglese con personale SHENKER più 2 ore al giorno di corso intensivo.
- Materiale audio-didattico per il Corso comprendente:
  * Walkmann
  * Cassette
  * Libri di testo e work book
- Assicurazione completa EUROPE-ASSISTANCE.
- Pernottamenti in hotel di categoria superiore e in Campeggi nei "NATIONAL PARKS".
- Tutte le tasse d'iscrizione. attrezzatura da pesca
  * canna da pesca con mulinello
  * licenza di pesca

I costi di noleggio del Super Van e tutti i costi annessi:
* benzina
* autostrade con pedaggio

Il programma si divide in 4 diversi turni a partire dall'ultima settimana di giugno
Prenotazioni e informazioni presso:

SHENKER INSTITUTE OF ENGLISH - Corso Monforte, 36 (MI) - Tel. 02/700332/700363/700929
ore ufficio - Sig.ra Sawchik - Olivieri (ore serali) Tel. 039/513211

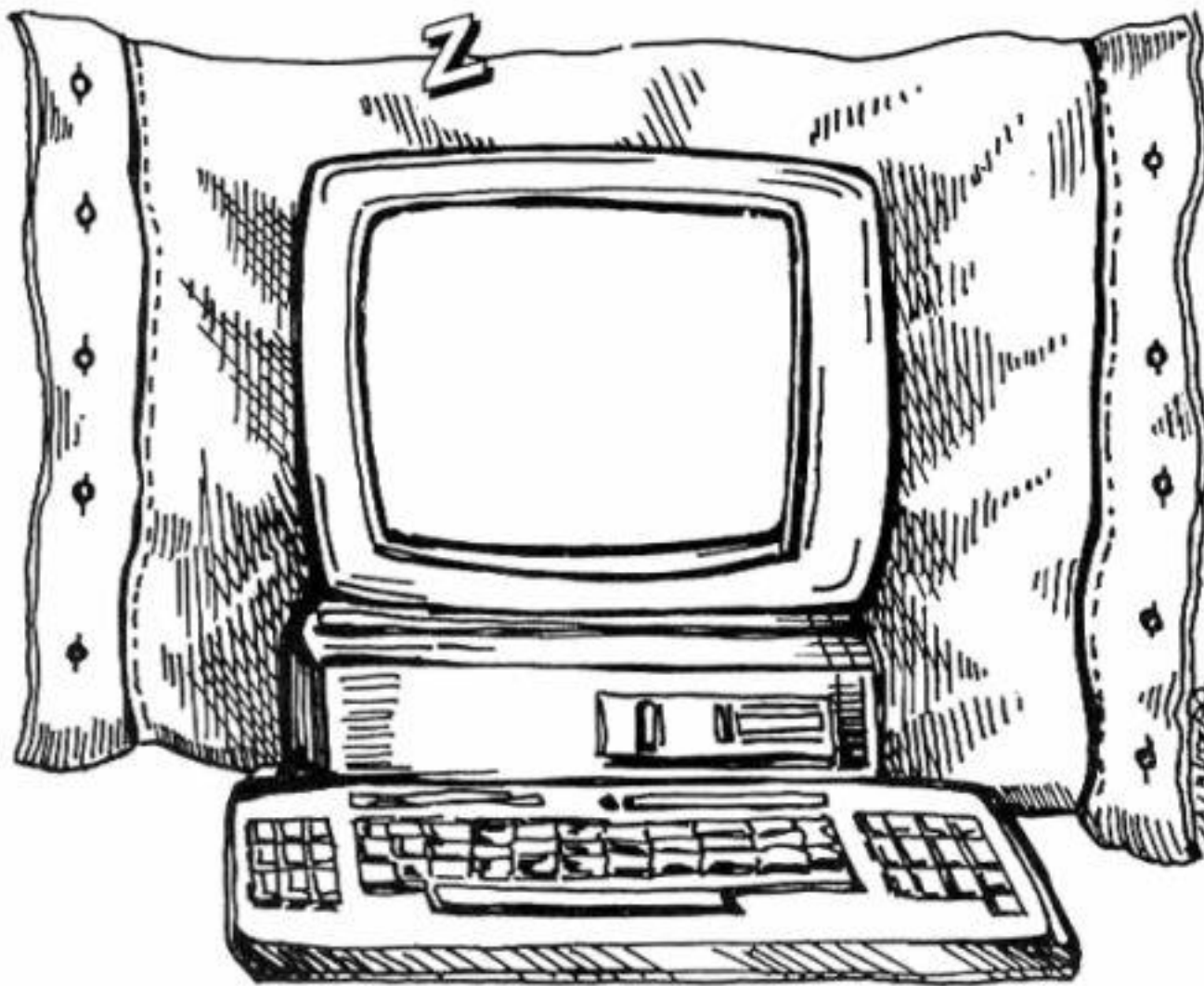
UVET - Viale Ferdinando di Savoia, 4 (MI) - Tel. 02/675061 (30 linee)
ore ufficio - Sig. Biagi

SYSTEMS - Viale Famagosta, 75 (MI) - Tel. 02/8467348/9
ore ufficio - Sig. Tidone

in collaborazione con:
SHENKER INST. OF ENGLISH
CP AIR
UVET

Salviamo il video

Un utile programma per costringere il C/64 a fare un po' di economia

di Michele Maggi

Chi usa il computer per molte ore al giorno, si sarà certamente reso conto che il più delle volte il monitor (o televisore) rimane inutilizzato.

Durante i lunghi minuti di "ragionamento", o durante le pause nelle quali il computer deve comunque rimanere acceso, il video consuma inutilmente sia i fosfori che l'elettricità.

Tempo fa, lavorando su di un terminale, ebbi l'occasione di notare che se la tastiera non veniva utilizzata per un certo periodo (alcune decine di secondi) il video si spegneva automaticamente.

Tale accorgimento, in sistemi che vengono utilizzati per svariate ore al giorno, comporta un notevole risparmio energetico e una minore usura del video. Anche molte calcolatrici tascabili, come è noto, si spengono automaticamente se dimenticate accese.

La routine proposta, che risolve il problema, è basata sulla programmazione in interrupt in modo da consentirne l'uso in abbinamento con altri programmi.

Il suo funzionamento è molto semplice e descritto nel commento del disassemblato; qualche nota esplicativa, comunque, non guasta.

Come funziona

Le quattro locazioni in pagina zero $02, $FB, $FC e $FD fungono da flag per il corretto funzionamento:

- $02 = flag di spegnimento: vale 1 se per il tempo stabilito non è stato toccato alcun tasto.
- $FB = primo flag di "non uso": viene incrementato ogni sessantesimo di secondo, se non viene premuto alcun tasto.
- $FC = secondo flag di "non uso": ogni volta che $FB supera 255 viene incrementato di 1 e comparato con Time e, se uguale, viene settato a 1 il flag di spegnimento.
- $FD = contiene il colore del bordo prima dello spegnimento.

Ad ogni interruzione il programma verifica se è settato il flag di spegnimento e, in caso affermativo, spegne il video e attende che sia premuto un tasto qualsiasi ad eccezione dei tasti Shift, Commodore, Control e Restore.

Come programmare il Timer

Per come è strutturato il programma è possibile stabilire un tempo minimo di circa 4.3 secondi e un tempo massimo di circa 18 minuti.

Ciò può essere definito da Basic cambiando il dato in linea 1080 (normalmente a 7) inserendo un valore compreso tra 1 e 255, oppure agendo direttamente con una Poke sulla locazione 53219.

Ogni unità vale circa 4.3 secondi; ne consegue che se il dato vale 1 il timer risulta settato a 4.3 secondi mentre, con il valore 7, il timer sarà settato a circa 30 secondi.

La routine è allocata a partire da 53000, in modo da interferire il meno possibile con altri programmi.

Nel caso, quindi, utilizziate altre routine, provvedete a rilocarlo altrove, in modo da averlo sempre a portata di mano.

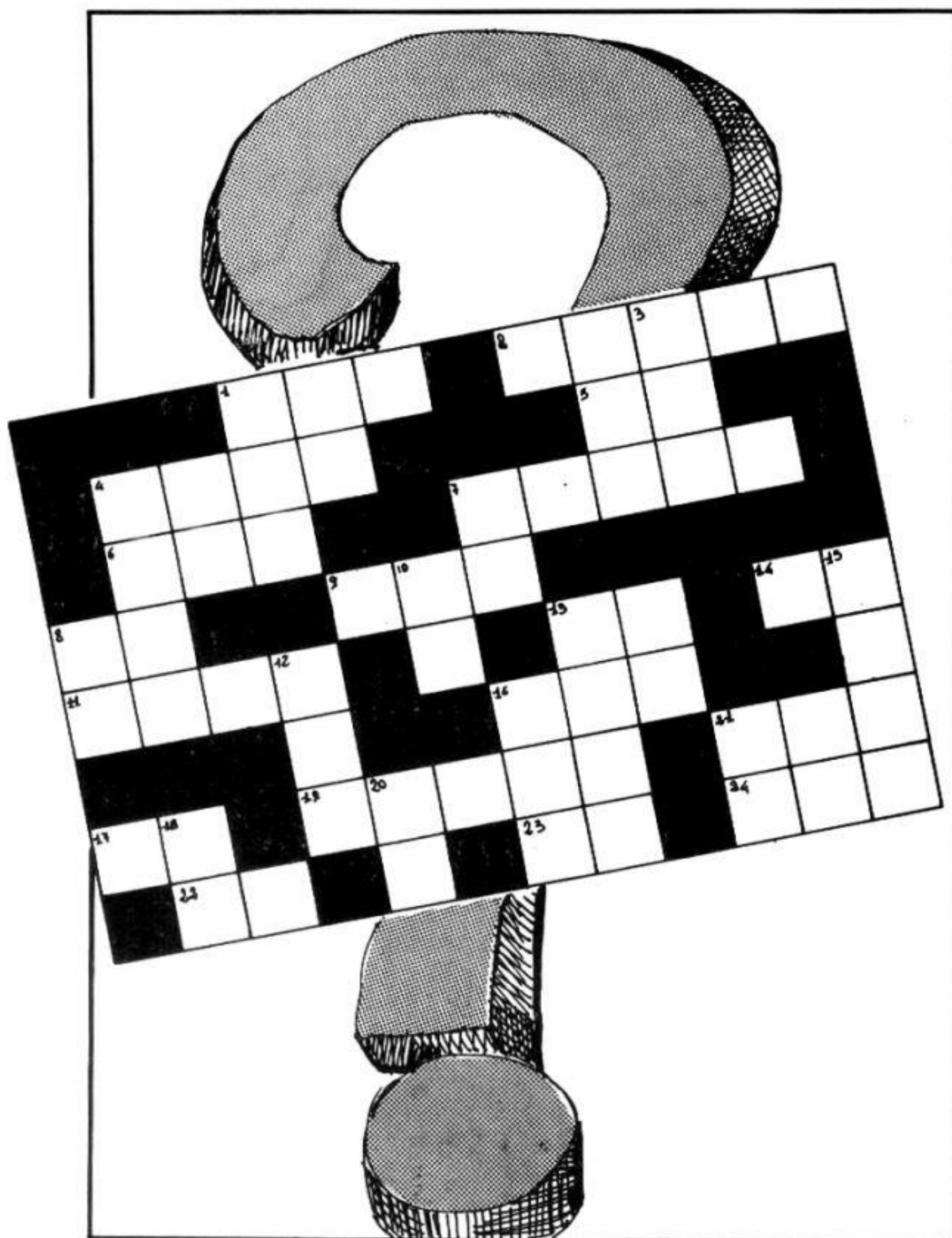
Nel caso in cui premiate Run/Stop e Restore, ovviamente, la routine viene disabilitata, seguendo le sorti di qualsiasi routine posta nell'Interrupt; un banale Sys 53100, naturalmente, provvederà a riattivarla.

```
1 *------------------------------*
2
3 * SALVAVIDEO PER C/64-128
4
5 * BY MICHELE MAGGI
6
7 * 11/5/87
8
9 *------------------------------*
10
11          LDA   #0         ;AZZERA
12          STA   $02        ;I FLAG
13          STA   $FB        ;DI
14          STA   $FC        ;CONTROLLO
15
16 ********************************
17
18          SEI              ;DIROTTA
19          LDA   #<ENTRY    ;I VETTORI
20          STA   $0314      ;DI INTERRUPT
21          LDA   #>ENTRY    ;FACENDOLI
22          STA   $0315      ;PUNTARE
23          CLI              ;A ENTRY
24          RTS              ;
25
26 ********************************
27
28 ENTRY    NOP              ;SE SI USA
29          LDA   $02        ;LA TASTIERA
30          BEQ   OLTRE      ;SALTA
31          LDA   #11        ;SE NO SPEGNE
32          STA   53265      ;IL VIDEO
33          LDA   $02        ;
34          BNE   PROX       ;
35          LDA   53280      ;SALVA IL
36          STA   $FD        ;COLORE IN $FB
37 PROX     LDA   #0         ;SCHERMO NERO
38          STA   53280      ;
39 LP00     LDA   $C5        ;SE VIENE
40          CMP   #64        ;PREMUTO UN
41          BNE   USA        ;TASTO SALTA
42          LDA   #1         ;SE NO SETTA
43          STA   $02        ;IL FLAG
44          JMP   $EA31      ;E VA A IRQ
45 OLTRE    NOP              ;RIPRISTINA
46          LDA   53280      ;IL COLORE
47          STA   $FD        ;DEL BORDO
48          LDA   $C5        ;SE VIENE
49          CMP   #64        ;PREMUTO UN
50          BNE   USA        ;TASTO SALTA
51          INC   $FB        ;SE NO INCREMENTA
52          LDA   $FB        ;IL PRIMO FLAG
53          BEQ   FLAG2      ;
54          JMP   $EA31      ;
55 FLAG2    INC   $FC        ;INCREMENTA
56          LDA   $FC        ;IL SECONDO FLAG
57          CMP   TIME       ;E LO COMPARA
58          BEQ   SPEGNE     ;SE E' UGUALE
59          JMP   $EA31      ;SE NO IRQ
60 SPEGNE   NOP              ;SETTA IL
61          INC   $02        ;FLAG
62          JMP   $EA31      ;E VA A IRQ
63 USA      NOP              ;RIAZZERA
64          LDA   #0         ;FLAG
65          STA   $02        ;DI
66          STA   $FB        ;CONTO
67          STA   $FC        ;E
68          LDA   #27        ;RIACCENDE
69          STA   53265      ;IL VIDEO
70          LDA   $FD        ;RIPRISTINA
71          STA   53280      ;IL COLORE
72          JMP   $EA31      ;E VA A IRQ
73 TIME     DFB   7          ;TIMER
```

```
10 REM *** SALVA-VIDEO ***
20 :
30 REM PER C/64
40 REM BY MICHELE MAGGI
50 REM (C) 1987
60 :
100 FORI=53100 TO 53218
110 READ A:POKEI,A
120 CK=CK+A:NEXT
130 READ A:POKEI,A
140 :
150 IF CK=16259 THEN170
160 PRINT"ERRORE NEI DATA":END
170 POKE53280,14:POKE53281,6:PR
    INTCHR$(154)
180 SYS 53100
190 TM=A*4.3
200 A$=LEFT$(STR$(TM),6)
210 PRINT"* SALVAVIDEO ATTIVATO
    *"
220 PRINT"TIMER SETTATO A"A$" S
    ECONDI"
230 :
1000 DATA 169,000,133,002,133,25
     1,133
1010 DATA 252,120,169,129,141,02
     0,003
1020 DATA 169,207,141,021,003,08
     8,096
1030 DATA 234,165,002,240,032,16
     9,011
1040 DATA 141,017,208,165,002,20
     8,005
1050 DATA 173,032,208,133,253,16
     9,000
1060 DATA 141,032,208,165,197,20
     1,064
1070 DATA 208,046,169,001,133,00
     2,076
1080 DATA 049,234,234,173,032,20
     8,133
1090 DATA 253,165,197,201,064,20
     8,027
1100 DATA 230,251,165,251,240,00
     3,076
1110 DATA 049,234,230,252,165,25
     2,205
1120 DATA 227,207,240,003,076,04
     9,234
1130 DATA 234,230,002,076,049,23
     4,234
1140 DATA 169,000,133,002,133,25
     1,133
1150 DATA 252,169,027,141,017,20
     8,165
1160 DATA 253,141,032,208,076,04
     9,234
1170 :
1180 DATA 007:REM RITARDO
```

Program Quiz

Ritieni davvero di essere bravo alla tastiera?

di Maurizio Dell'Abate

Pubblichiamo in queste pagine un gruppo di semplici programmi: il vostro compito è individuarne il risultato, scegliendone uno fra quelli proposti.

Se proprio non ce la fate, siete autorizzati a consultare il computer; le risposte corrette, infatti, non le forniamo proprio per stimolare la vostra curiosità...

Alcuni di questi programmi possono essere utilizzati così come sono divenendo quasi utili (!), altri devono essere corretti perchè pieni di orribili errori, altri ancora non servono assolutamente a nulla.

#1

```
10 PRINT "PIPPO"
20 PRINTCHR$(147)
30 PRINT "COMMODORE"
40 GOTO30
```

a) stampa "PIPPO"

b) stampa "PIPPO" e "COMMODORE", alternativamente

c) stampa innumerevoli "COMMODORE"

#2

```
10 INPUTA
20 X=SQR(3)/2
30 Y=A*X
40 Z=A*6
50 A=Z*Y/2
60 PRINTA
```

a) serve per calcolare l'area di un esagono regolare di lato A

b) serve per calcolare il lato di un dodecagono regolare di area A

c) non serve a nulla, anzi si blocca con un errore nella riga 20

#3

```
10 FORA=1TO10
20 GOSUB100
```

```
30 PRINTA,A$(A)
40 NEXT
50 DIMA$(13)
60 FORA=1TO13
70 GOSUB100
80 PRINTA,A$(A)
90 NEXT:END
100 X=RND(0)
110 IFX<.4THENA$(A)= "1":RETURN
120 IFX<.9THENA$(A)= "X":RETURN
130 A$(A)= "2":RETURN
```

a) stampa una schedina casuale del totocalcio

b) inizia a far qualcosa... ma non finisce

c) riempie lo schermo di "1", di "X" e di "2"

#4

```
10 INPUT "ALTEZZA CM (>100)";A
20 INPUT "PESO KG";P
30 PF=A-100
40 PRINT"RISULTATO ";P-PF
50 IFP-PFTHENEND
60 PRINT"PESO PERFETTO!"
```

a) calcola il peso forma e indica i kg in eccesso (+) o in difetto (-)

b) funziona in parte perchè è presente un errore logico di programmazione

c) si blocca con un errore nella riga 50

#5

```
10 INPUTA$
20 FORI=1TOLEN(A$)
30 PRINTVAL(A$)*I
40 NEXT
```

a) non è di alcuna utilità pratica

b) si blocca con un errore nella riga 30

c) stampa la tabellina del numero digitato in fase di Input

#6

```
10 INPUTA
20 HB=INT(A/255)
30 LB=A-HB*255
40 PRINTLB,HB
```

a) stampa byte basso e byte basso del valore impostato
b) stampa due valori senza significato
c) stampa due sottomultipli del valore impostato

#7

```
10 A=ASC("NBA BASKET")
20 A$= "DISK DRIVE: DEVI
COMPRARLO"
30 X$=LEFT$(A$,10)
40 G$=LEFT$(X$,LEN(X$)+A-ABS(A))
50 OPEN8,8,15,CHR$(A+1-2+1)+CHR$
(ASC(":C:C:C:"))+G$:CLOSE8
```

a) cancella il dischetto presente nel drive e assegna il nome "NBA BASKET" al floppy
b) lo formatta, ma con il nome "DISK DRIVE: DEVI..."
c) cancella il floppy e assegna allo stesso il nome "DISK DRIVE"

#8

```
10 A=49152:REM LOCAZIONE LIBE-
RA (C-64)
20 POKEA,96
30 A$= "SELF BREAK MICRO"
40 SYSA:A$= "CHANNEL"
50 SYSA:A$= "ERROR"
60 SYSABS(A)+SGN(A)-SIN(LOG(3)/2)
70 SYSA:STOP
```

a) non genera risultati evidenti
b) inchioda irreparabilmente il microprocessore
c) danneggia la tastiera

#9

```
10 INPUTA
20 INPUTB
30 A%=A
40 B%=B:B=ABS(B):B%=B:B=B%:B%=
B%
50 A=A*100
60 B=B*A/A%
70 PRINTA%,B%
```

a) stampa due valori che non hanno evidente legame con i due immessi
b) stampa la percentuale di A e di B
c) stampa i numeri immessi senza parte decimale

#10

```
10 THE=1
20 PRIN=2
30 GRAPHI=3
40 PRINTSTR$(THE)+STR$(PR)+
STR$(G)
```

a) si blocca con un errore nella riga 10
b) stampa " 3"
c) stampa " 6"
d) stampa "120" (separati da spazi)
e) stampa "123" (senza spazi tra loro)
f) non stampa nulla
g) si blocca con un errore di sintassi

#11

```
10 INPUTA
20 INPUTB
30 X=A↑(1/B)
40 PRINTX
```

a) stampa un numero che non ha particolari legami con i due valori immessi

b) stampa la radice di A con indice B

c) emette un errore nella riga 30

d) stampa la cotangente iperbolica inversa del numero A tenendo conto del fattore di riduzione B

e) si fonde il microprocessore

f) vengono smagnetizzati tutti i dischetti presenti nel raggio di 3 metri

#12

```
10 GOSUB30
20 GOSUB40
30 GOSUB20
40 GOSUB10
```

a) viene creato un ciclo infinito (run/-stop per interrompere)
b) si blocca con un errore nella riga 20
c) si ferma dopo circa 15 secondi
d) si ferma dopo circa 16 minuti e colora lo schermo in bianco
e) fa squillare il telefono rosso della Casa Bianca
f) dodici sfitinzie bussano alla vostra porta.

Directory

Tutti i programmi pubblicati su questo numero di Commodore Computer Club, sono registrati su un dischetto appartenente alla serie "Directory".

Oltre ai programmi citati, sono presenti altri file di notevole interesse per coloro che desiderano realmente utilizzare il proprio computer.

Sono infatti presenti, di solito, anche i programmi pubblicati sull'altra nostra rivista "Personal Computer", ed altri file che, in totale, riempono quasi per intero i 664 blocchi normalmente disponibili su un floppy disk.

Sono disponibili i seguenti dischetti:

***Directory N. 1** (CCC N.34 + raccolta dell'intera Enciclopedia di routine)*
***Directory N. 2** (CCC N.35 + CCC N.36)*
***Directory N. 3** (CCC N.37 + PC N.7 + PC N.8)*
***Directory N. 4** (CCC N.38 + file vari)*
***Directory N. 5** (CCC N.39 + PC N.9)*
***Directory N. 6** (CCC N.40 + PC N.10)*
***Directory N. 7** (CCC N.41 + file vari)*
***Directory N. 8** (CCC N.42 + file vari)*
***Directory N. 9** (CCC N.43 + file vari)*

Come procurarsi i dischetti della serie "Directory"

Avvertiamo i lettori che NON è assolutamente possibile inviare i programmi su nastro, per intuibili motivi di economia ed affidabilità del nastro cassetta.

Ogni numero di "Directory" può quindi esser richiesto SOLO su disco inviando L.12000 per ciascun disco oltre a L.3000 (fisse) per le spese di imballo e spedizione (indipendenti dal numero di dischi richiesti).

Chi desiderasse la spedizione raccomandata, deve aggiungere altre 3000 lire per l'ulteriore affrancatura.

Non ci è possibile inviare materiale contrassegno: si prega di astenersi dal chiedere eccezioni alla regola.

Compilate un normale modulo di C/C postale indirizzando a:

C/C postale N. 37952207
Systems Editoriale
Viale Famagosta, 75
20142 Milano

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo, ma anche il numero del disco desiderato; esempio:

"Directory N.1"
"Directory N.3"
"Directory N.4"

Totale:
L.12000x3 +
L.6000 (spediz. racc.)
= L.42000

(spese di imballo e spediz. racc. comprese).

N.B. Per ottenere il materiale ordinato in tempi più ristretti, inviate l'importo a mezzo assegno bancario non trasferibile con lettera di accompagnamento: le poste italiane non brillano per velocità! (due mesi circa per il recapito di un C/C postale).

Amiga 2000, l'evoluzione della specie

Il nuovo "mostro" Commodore sembra proprio confermare tutte le aspettative: bello, affidabile e potente: di più proprio non si può...

di Michele Maggi

A poco più di un anno dalla presentazione della prima serie Amiga, eccone una versione più potente, senz'altro migliorata e pronta per soddisfare anche l'utente più esigente.

Esattamente un anno fa ebbi l'onore (e l'onere!) di scrivere il primo articolo di C.C.C. dedicato all'Amiga.

Fra l'enorme quantità di articoli che furono scritti, per l'occasione, sulle varie riviste specializzate, quello da noi pubblicato attirò l'attenzione non per l'argomento Amiga in se stesso, ma perchè fu forse l'unico a parlare dell'Amiga con la dovuta obiettività e non con i toni deliranti propri di periodici religiosi.

La filosofia della nostra rivista, infatti, è quella di informare correttamente il lettore e di fornirgli, quindi, il maggior numero di notizie "reali" (e utili) e non "sparate", tipiche di volantini e slogan pubblicitari.

Va da sè che l'articolo citato provocò numerose critiche da parte, guarda caso, dei rivenditori, seccati per evidenti motivi.

Parallelamente alle lettere di critica giunsero anche parecchie lettere (questa volta di lettori) che ci incoraggiavano a proseguire fornendo guidizi obiettivi sia sull'hardware che sul software.

Il resto è noto: la prima versione di Amiga 1000 qualche piccolo (!) problema l'aveva davvero ma, comunque, due successive release di Kickstart e Workbench sembrano aver rimesso tutto a posto.

Un altro particolare che nessuno degli attenti critici si è preoccupato di notare è che, al tempo della prova, l'Amiga costava la bellezza di tre milioni mentre adesso lo stesso Amiga

AMIGA
Commodore MODEL 1081
POWER
Commodore
AMIGA 2000

costa, in alcuni negozi, una cifra che si avvicina molto alla metà del prezzo iniziale.

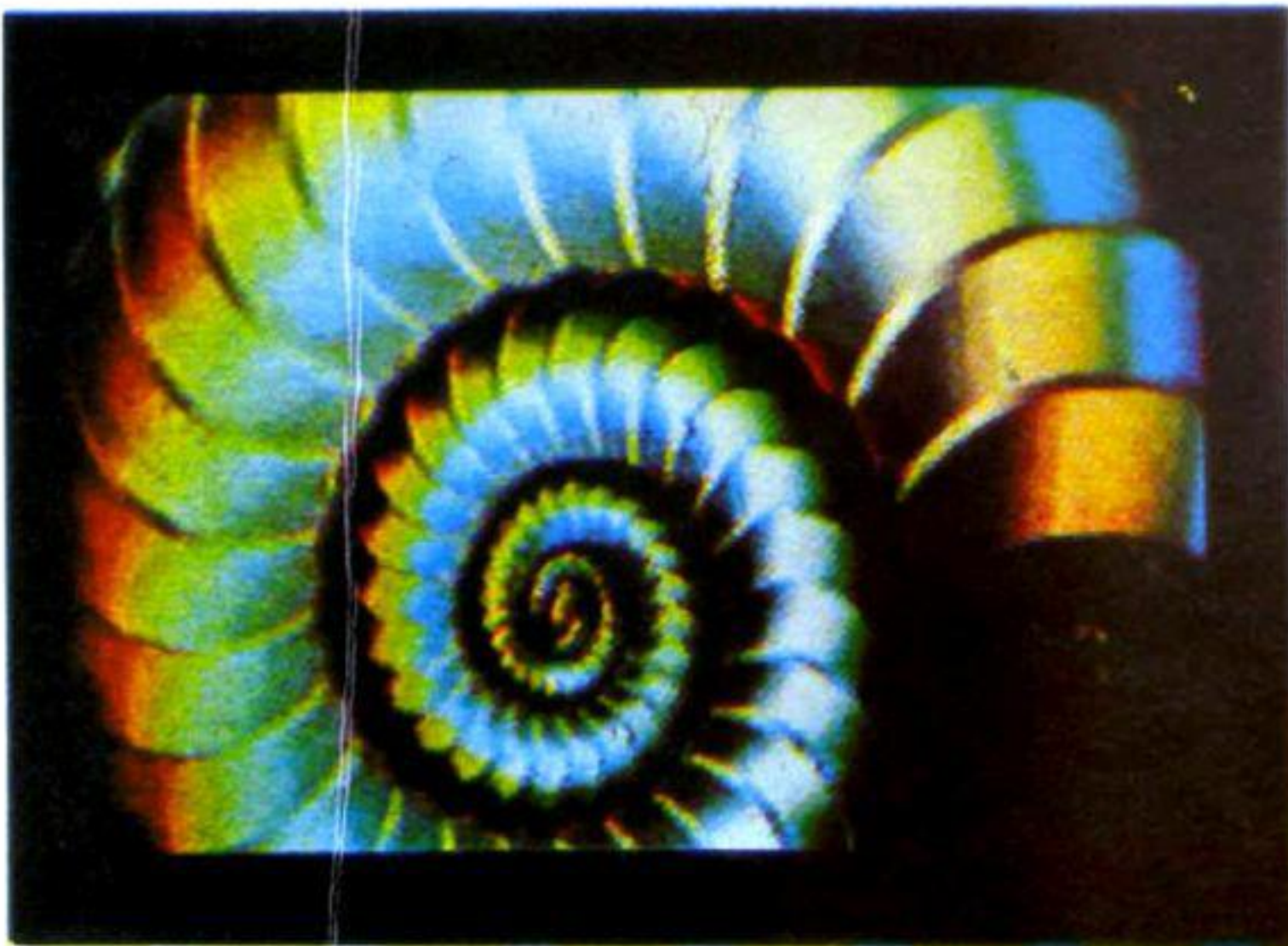
Non occorre essere geni della finanza per comprendere che ciò modifica di molto il rapporto qualità prezzo...

Ad un anno di distanza la Commodore, indubbiamente cresciuta e molto più attenta alle esigenze del mercato, presenta l'Amiga 2000: Kickstart in Rom, un Megabyte di memoria interno, reali possibilità di espansione (Hard disk, Drive 5 1/4, fino ad 8 Mega esterni, scheda compatibilità IBM).

La versione del 2000 provata è quella "base", vale a dire dotata di un solo drive da 3.5 e senza espansioni di memoria.

La configurazione base permette già di lavorare ad un buon livello, in ambiente Amiga Dos, ma per un utilizzo professionale è raccomandabile, quantomeno, l'acquisto di un secondo drive da 3.5.

Per ciò che concerne il Workbench (1.2) possiamo notare una "dotazione" migliorata rispetto a quella del Workbench precedente, anche se le modalità d'uso sono le solite.

Per quanto riguarda le prestazioni solite, tipiche dell'Amiga (suono, grafica ed altro) nulla da dire che già non sia stato detto: suona e disegna fin troppo bene, specialmente se a farlo disegnare, o cantare, sono programmi di alto livello come Aegis Animation e Sonix.

La morale è la solita: qualsiasi computer, anche il migliore, ha tassativamente bisogno di buon software e, nel caso Amiga, beh... non ci si può proprio lamentare!

Anzi, a proposito di buon software, chi non è rimasto allibito dalla stupenda schermata grafica di "The Pawn" e soprattutto dalla sua meravigliosa musica?

Molto del software per Amiga attualmente in circolazione (già piratеggiato, in accordo con un mal(?)costume tradizionale che si va consolidando sempre più) dimostra che anche le maggiori Software-House sono coinvolte nel fenomeno Amiga e sono costantemente al lavoro per creare programmi di alto livello.

Un chiaro esempio può essere visto in Aegis Sonix, un programma musicale che, oltre a suonare stupendamente (con suoni campionati!) dimostra, tramite una magistrale gestione del multitasking, come sia

C.A.T.M.E.

CENTRO ASSISTENZA TECNICA MINI - ELABORATORI

Via F. Severoli, 9 - 20147 MILANO
Via L. Cavalieri, 3 - 20147 MILANO
Tel. (02) 4158638 - 4152962

IL CENTRO AUTORIZZATO PER TUTTI I PRODOTTI COMMODORE

PER SERVIZIO ASSISTENZA IN GARANZIA E NON

- *ricambi originali*
- *servizi qualitativi ed estremamente rapidi*
- *contratti manutenzione on-site*
- *possibilità di prenotare giorno ed ora per la più sollecita riparazione dell'apparecchiatura*
- *prese e consegne su tutto il territorio*
- *consulenza tecnica hardware*

PARTICOLARI CONDIZIONI PER RIVENDITORI E DISTRIBUTORI

- *VENDITA AMPIA GAMMA ACCESSORI*
- **APERTO IL SABATO MATTINA**

Consegnando il tagliando a lato, con l'apparecchio da riparare, avrete diritto:
- *Alla riparazione in garanzia totalmente gratuita + omaggio*
- *Alla riparazione fuori garanzia con sconto 20% sulle tariffe Commodore*

possibile sfruttare a fondo tutte le potenzialità della macchina.

E' possibile gestire e modificare uno spartito in tempo reale (=mentre sta suonando) senza che il funzionamento del programma ne risenta minimamente; è anche possibile creare e modificare le forme d'onda dei vari suoni e "fare tutte quelle cose" che fanno letteralmente impazzire gli appassionati di musica (elettronica e non).

La nota dolente

Torniamo ora al tema iniziale, ovvero splendori e miserie dell'Amiga 2000; niente paura per gli affamati di novità software: d'ora in poi dedicheremo alle più interessanti novità uno spazio fisso in modo da soddisfare anche questa fascia di lettori.

Amiga 2000: splendori tanti e miserie poche, molto poche; l'unico problema che realmente può comportare inconvenienti è il seguente: alcuni programmi (pochi) che girano perfettamente su Amiga 1000 e Amiga 500 non girano sul 2000. Come mai? Di chi è la colpa?

Ecco per tutti la risposta a questi angosciosi interrogativi che hanno tolto il sonno per parecchio tempo al sottoscritto e agli altri (s)fortunati redattori che hanno collaborato ai test della macchina.

Andiamo con ordine, la macchina è la stessa, i programmi gli stessi, ma... allora qual'è il problema? Un Megabyte invece di 512K, ecco la risposta!

Evidentemente chi ha realizzato i programmi in questione, facendo largo uso di routine del Sistema Operativo, non ha utilizzato gli "Entry Point" legali, cioè quelli ufficialmente comunicati dalla Commodore.

Che cos'è un Entry Point?

Chiarisco subito questo concetto con un esempio di facile comprensione per i "sessantaquattristi": la routine del Kernal "Chrout" dei Commodore a 8 bit (64, 16, 128) è allocata da $FFD2 in poi; chi intenderà utilizzarla, quindi, dovrà farlo "chiamandola" al suddetto indirizzo mediante una Sys oppure un opportuno programma in linguaggio macchina.

Supponendo, però, di conoscere a fondo il S.O. possiamo permetterci di entrare in un punto "illegale" della suddetta routine per scopi particolari, tra cui, per esempio, una maggior velocità di esecuzione ottenibile

bypassando alcuni controlli ritenuti superflui.

E' intuitivo che se la Commodore riscrivesse (in tutto o in parte) il S.O. del C/64, lasciando però invariati gli Entry Point delle versioni precedenti, i programmi che al loro interno hanno delle chiamate ad Entry Point illegali rischiano di non funzionare correttamente: ed è proprio ciò che accade, in alcuni casi, utilizzando alcune versioni velocizzatrici (Speed Dos, Prologic ed altre).

Ecco dunque spiegato il mistero: evidentemente il S.O. del 2000, pur essendo in parte differente dalla versione precedente, mantiene inalterati gli entry point, ma ciò non è sufficiente per alcuni programmi scritti con disinvoltura, e ciò genera i suddetti inconvenienti.

L'Amiga 2000, quindi, funziona perfettamente: si tratta di avere solo un po' di pazienza e attendere l'uscita della nuova release dei programmi incriminati.

Ad un anno di distanza dall'ingresso nel mercato dell'Amiga, l'opinione che si può trarre è positiva, sia nei confronti della macchina, sia nei confronti della stessa Commodore, che sta dimostrando di occuparsi di informatica professionale e non solo di home computer e videogame.

LA PERFEZIONE DIVENTA MITO

MITO - 5 1/4" Floppy 48 TPI
Doppia Faccia - Doppia Densita
Garantito al 100% - Velocità di
registrazione 5800 BPI
600.000 bytes unformatted.

le misure della perfezione

RECOVERY SERVICE - Un nostro servizio esclusivo. Cosa è il Recovery Service? È uno scudo a protezione del vostro lavoro. Se per un incidente qualsiasi: macchie di caffè, di cioccolato o impronte, il vostro disk dovesse danneggiarsi, la MICROFORUM è in grado di recuperare i dati senza alcun esborso da parte vostra.

La *MICROFORUM MANUFACTURING INC.* è interessata all'ampliamento della propria rete distributiva. Per qualsiasi contatto scrivere anche in italiano.

944-A St. Claire Ave. West TORONTO, CANADA M6C 1C8 Tel. (416) 656-6406 - Tlx. 06-23303 MICROFORUM TOR. Telefax (416) 656-6368

STUDIO C SANTORI Roma

Parlar d'Amiga

Alcune informazioni sull'uso corretto del sintetizzatore vocale del potente computer

di Luigi Callegari

Come molti sapranno, Amiga dispone di un completo e sofisticato digitalizzatore vocale in grado di riprodurre con notevole fedeltà la voce umana.
Vi si può accedere per molte vie, variabili a seconda della versione di Workbench posseduta:

- Da Workbench: selezionando "System" e poi "Say"
- Da Basic, usando il dischetto "Extras" e selezionando il programma "Speech".
- Da CLI, premendo il tasto Return dopo aver digitato...:

SAY

...oppure...:

RUN System/Say

...quest'ultimo per le versioni più vecchie di Workbench.

Il modo più semplice per familiarizzare con il sintetizzatore sonoro è quello di usare il programma Basic, che ne consente un uso semplificato con il mouse.

Da CLI è possibile usare direttamente un comando del tipo:

SAY "Hello, How are you"

mentre, digitando semplicemente...:

SAY

...si entra in un modo interattivo grazie alla comparsa di due finestre (window): nella "Input Window" si digitano direttamente la parole (inglesi) da riprodurre, e nella "Phone-

LA PERFEZIONE DIVENTA MITO

QUAD-MITO - 5 1/4" 96 TPI DS/QD
Floppy disk a quadrupla densità, disegnato per aumentare la capacità di registrazione sino a 780 kb per dischetto.
Velocità di registrazione 5800 BPI

MEGA-MITO - 5 1/4" 96 TPI HIGH DENSITY
Floppy ad alta densità, disegnato per drive da 1.2 MEG (AT e compatibili).
Velocità di registrazione 9650 BPI

MICRO-MITO - 3 1/2" 135 TPI DS/DD
Costruito per l'era dei disk drive da 3 1/2".
Velocità di registrazione 8100 BPI

le misure della perfezione

944/A St. Claire Ave. West TORONTO. CANADA M6C 1C8
Tel. (416) 656-6406 - Tlx. 06-23303 MICROFORUM TOR
Telefax (416) 656-6368

La ***MICROFORUM MANUFACTURING INC.***
è interessata all'ampliamento della propria rete distributiva.
Per qualsiasi contatto scrivere anche in italiano.

STUDIO C.SANTORI Roma

me Window" appare la conversione effettuata dal sistema nel codice a fonemi, mentre dall'altoparlante viene riprodotta la voce.

E' sempre possibile controllare l'emissione della voce specificando alcune opzioni. Ciò si ottiene inserendo, nella stringa argomento di SAY, un segno di meno (-) seguito da una lettera e, se necessario, da un parametro numerico.
Le opzioni sono le seguenti:

-n :impone di usare una voce naturale
-f :impone di usare una voce femminile (più acuta)
-m :impone di usare una voce maschile (più grave)
-r :impone di usare una voce robotica (monotona)
-pXXX :fissa al valore XXX la frequenza della voce
-sYYY :fissa al valore YYY la velocità della voce

Il parametro numerico per la frequenza (pitch) può variare tra 65 e 320 (il default è 110 per una voce "media" maschile); quello per la velocità è compreso tra 40 e 400 (il default è 150).
Si provi ad esempio da CLI (in modo interattivo si può tralasciare la specifica SAY):

SAY -f -p250 -s130 The Amiga can talk like a nice female

SAY -r -p90 This is a robotic voice

SAY -n -s40 I am so tired

I parametri vengono considerati costanti per i comandi successivi, sinchè non si alterano con un nuovo parametro.
Ovviamente il modulo di traduzione delle parole è predisposto per la lingua inglese, ma sembra che in un prossimo (!) futuro la Commodore distribuirà il file necessario per codificare in fonemi e quindi pronunziare parole italiane.
Da Basic è comunque possibile controllare anche altri parametri per la sintesi vocale (sono nove in tutto). Chi conosce l'inglese può esaminare il manuale originale di AmigaBasic alla sezione 8-130 dedicata ai comandi Translate$ e Say.
Il primo comando converte in fonemi la stringa di argomento, mentre SAY pronuncia i fonemi stessi.

Ad esempio si può inserire in un listato, oppure in modo diretto:

SAY TRANSLATE$("This is my voice")

In questo caso l'elaboratore chiederà l'immissione del disco di Workbench per caricare il modulo di traduzione (presente nella directory DEVS) necessario alla conversione in fonemi.

Oppure si può specificare la stringa, o variabile alfanumerica, seguita da una matrice contenente, in successione, i nove parametri del comando SAY secondo un formato tipo:

SAY TRANSLATE$ ("stringa"), matrice%

La "matrice" di valori interi deve contenere, ovviamente, nove parametri, numerati da zero a otto (secondo gli indici della matrice), con i seguenti significati:

0) Frequenza base di pronuncia in Hertz (65-320), Default=110.
1) Modulazione. 0=Enfasi e inflessione attive. 1=monotono.

2) Velocità in parole al minuto (40-400). Default=150.
3) Sesso della voce. 0=maschile. 1=femminile. Default=0.
4) Frequenza di campionamento (5000-28000). Default=22200.
5) Volume (0-64). Default=64.

6) Canale (0-11). Default=10. I canali 0 e 3 sono a sinistra, 1 e 2 a destra; approfondiremo l'argomento più avanti.
7) Modo (0-1). 0=arresto esecuzione programma sino al termine della pronuncia della parola. 1=continua esecuzione dopo avere iniziata la pronuncia (modo asincrono).
8) Controllo (0-2). Indica al sistema come comportarsi durante il processo asincrono di più specifiche SAY (solo se il parametro precedente è fissato a uno):
0= (default) finisce il primo SAY incontrato e poi esegue il successivo.
1= cancella il comando precedente
2= interrompe subito il primo SAY ed inizia il secondo.

Il settimo parametro, memorizzato cioè nella sesta posizione della matrice, fissa quali dei quattro canali (0-3) di uscita di Amiga riceve il segnale del sintetizzatore vocale. Poichè i quattro canali vengono miscelati in due canali stereo (sinistro e destro) questo parametro fissa anche da che parte è riprodotto il segnale sonoro:

Val.	Canale
0	0 (solo verso l'altoparlante sinistro)
1	1 (solo verso l'altoparlante destro)
2	2 (solo verso l'altoparlante destro)
3	3 (solo verso l'altoparlante sinistro)
4	0 e 1 (ambedue gli altoparlanti)
5	0 e 2 (ambedue gli altoparlanti)
6	1 e 3 (ambedue gli altoparlanti)
7	2 e 3 (ambedue gli altoparlanti)
8	0 e 3 (altoparlante sinistro)
9	1 o 2 (altoparlante destro)
10	Qualunque combinazione libera (destro e sinistro)
11	0, 1, 2 o 3 (destro o sinistro)

I valori da zero a sette forzano la generazione verso il/i canale/i specificati e, più precisamente, 0-3 per inviare verso un singolo canale e 4-7 per una coppia di canali.
Il valore otto consente al sistema di scegliere un canale libero tra i due che inviano all'altoparlante sinistro il segnale, mentre il valore nove esegue la stessa funzione per il canale destro.

Il valore dieci, che è di default, lascia scegliere al sistema la combinazione di canali liberi a sinistra ed a destra ma presuppone almeno due canali liberi. Il valore undici è indispensabile per casi limite, perchè forza l'invio del segnale ad un qualunque singolo canale libero.

Cronaca di un'Amiga annunciata

Considerazioni a ruota libera sui recenti modelli di marca Commodore

di Alessandro de Simone

Ne vale la pena?

L'Amiga è un computer dotato di 96 tasti e quindi è più potente di uno che ne ha 83.

No, così non va: è assurdo presentare un computer mediante confronti stupidi; allora tentiamo in modo diverso:

L'Amiga ha un microprocessore 68000 che viaggia a 7 megahertz. Forse così va meglio, ma chi non se ne intende non ci capisce un'acca; chi se ne intende, invece, non può accontentarsi di un'informazione più che nota.

Un'occhiata alle altre riviste del settore, che hanno presentato l'Amiga in precedenza, non mi ispirano granchè: tutte si affannano a contare il numero dei piedini dei chip, parlano di compatibilità con l'IBM (molte senza aver visto in funzione nemmeno il sidecar), osannano la notevole memoria e, in certi casi, si auto-pavoneggiano in deliranti editoriali farciti di "eccezionale", "stupendo", "meraviglioso" e simili.

Da oltre un anno, ormai, bazzico con l'Amiga, ma solo ora decido di parlarne in maniera sistematica.

Ma ciò che intendo porre in evidenza in queste pagine è soprattutto una "procedura" per tentare di dare la risposta ad una domanda che molti sessantaquattristi sospettosi si posero quando l'Amiga fu presentato ufficialmente:

"Vale la pena passare dal C/64 all'Amiga?"

E' necessaria una "carrellata" sulla storia dell'informatica di massa (in generale) e Commodore (in particolare). Solo in questo modo, infatti, è possibile inquadrare correttamente, a mio parere, il fenomeno Amiga.

Un po' di storia

Il Vic-20, capostipite vero dell'informatica in famiglia, rappresentò un successo clamoroso: l'enorme convenienza del prezzo, la possibilità di collegarsi ad un TV domestico (che evitava, quindi, l'acquisto di un monitor dedicato), la capacità sonora e l'uscita a colori (impensabili, a quei tempi, in computer di basso costo), ponevano infatti in secondo piano la minima dotazione di memoria Ram (solo 3.5K) e la dimensione dello schermo (22 righe x 23 colonne).

Il Vic-20 insegnò (ma non tutti appresero le sue lezioni) che era possibile dotare ogni famiglia di una tastiera, purchè questa riuscisse a stupirla e a divertirla.

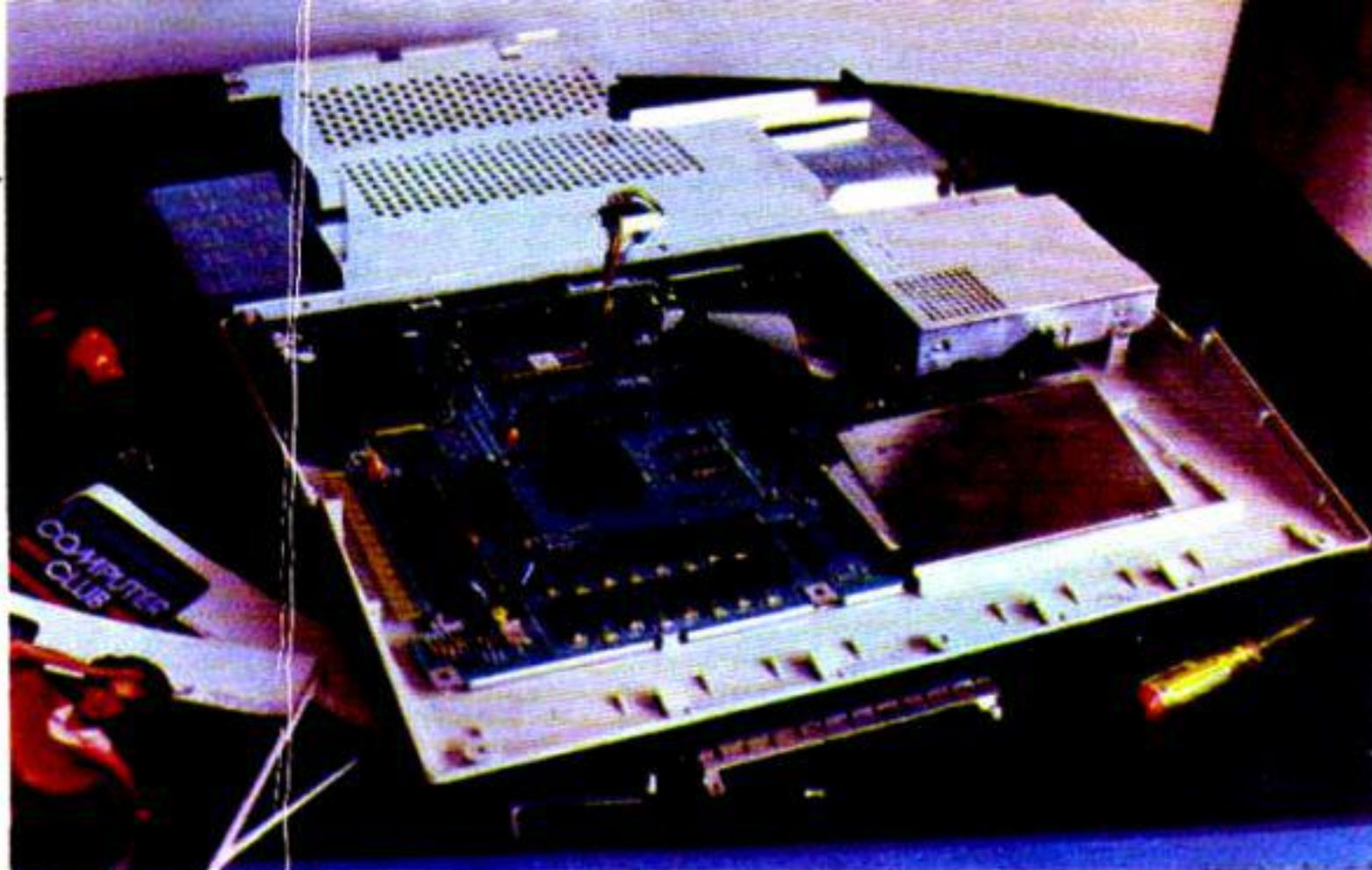
Il passo successivo della Commodore fu rappresentato dal C/64.

Quando mi fu consegnato il primo esemplare giunto in Italia, allo scopo di presentarlo ai grossisti (durante una delle prime presentazioni ufficiali dell'allora neonata Commodore Italia), compresi che quel computer aveva tutti i numeri per un roseo futuro: la dotazione di Ram incredibile (il calcolatore più dotato di quei tempi raggiungeva solo i 32K), il colore, il suono (tuttora insuperato) e gli sprite, aprirono le porte ad una diffusione indiscriminata di applicazioni, utility, giochi e package professionali che ancora oggi non sembra essersi esaurita.

Il motivo per cui il C/64 sbaragliò il Vic 20 era rappresentato dal fatto che era migliore *sotto tutti i punti di vista.* Il Vic-20, espanso al massimo, costava, grossomodo, come un C/64 e nessuno si sognò di acquistare più il Vic-20, neanche quando, per motivi di politica commerciale, questo fu posto in vendita ad un prezzo inferiore alle 200 mila lire.

La disattenzione con cui questo dato, a mio parere, fondamentale, fu valutato, portò la Commodore ad imbarcarsi in successivi esperimenti produttivi, non proprio felici.

Il C/16 avrebbe dovuto rappresentare, infatti, il computer dal prezzo stracciato (meno di 200 mila lire anch'esso) che permetteva, a differenza del C/64, una gestione della grafica e del suono decisamente facilitata. Ed in effetti il C/16 rispettava ampiamente tale caratteristica: chiunque abbia avuto a che fare con questo computer si è divertito moltissimo a disegnare e a riprodurre effetti sonori con la semplicità tipica del linguaggio Basic.

Purtroppo non furono presi in esame alcuni aspetti notevoli, che impedirono, di fatto, il decollo del piccolo computer: anzitutto, il C/16 veniva proposto a distanza di tempo dal C/64; e nel frattempo molte software house avevano provveduto a diffondere a macchia d'olio utility e linguaggi, per il C/64, che supplivano alle carenze del Basic V.2 di cui era dotato. Possiamo affermare, anzi, che uno dei successi del C/64 era proprio la notevole "apertura" del sistema e la conseguente completa libertà di azione tanto per le software house quanto per gli utenti finali stessi.

Privato di una caratteristica che, ormai, era propria anche del C/64, il piccolo C/16 evidenziava altri talloni d'Achille: la quantità modesta di memoria Ram, che impediva la stesura di programmi di ampio respiro (l'entrata in alta risoluzione assorbiva, da sola, 8 KRam), la mancanza di sprite (che limitava la realizzazione di animazioni spettacolari), la mancanza del SID, il chip sonoro del C/64, che consentiva effetti sonori degni di nota.

Fu trascurata la lezione impartita dal Vic 20 che aveva dimostrato, senza ombra di dubbio, che la gente spende volentieri una cifra maggiore pur di procurarsi un'attrezzatura che possa garantire adeguate soddisfazioni.

Il numero di utenti del C/16, quindi, risultò piccolo con la conseguenza di spingere le software house a disertarlo per dedicarsi alla realizzazione di software, più produttivo, per il solo C/64. A chi poteva convenire, infatti, dedicare del tempo per scrivere un videogioco di cui si sarebbero vendute poche copie? Era ovviamente più utile scrivere programmi per un potenziale parco utenza decisamente più vasto: quello del C/64.

Al Plus-4 era riservata una sorte, se possibile, peggiore: la Ram era delle dimensioni adatte alla classe del computer, ma i tre programmi incorporati risultavano deludenti; ci si accorse in fretta che un C/64, dotato di tre programmi su dischetto, era nettamente più versatile e professionale di un Plus-4, nonostante la comodità di disporre di tre programmi all'accensione della macchina.

A causa del settore "professionale" cui si era rivolta la Commodore per la vendita del Plus-4, inoltre, scarso era l'interesse per la produzione di software di intrattenimento.

Il C/128, quindi, sembrava rappresentare la giusta soluzione di compromesso ed avrebbe avuto tutte le carte in regola per proporsi come successore del C/64: prezzo leggermente più elevato ma potenzialità nettamente maggiori rispetto al C/64.

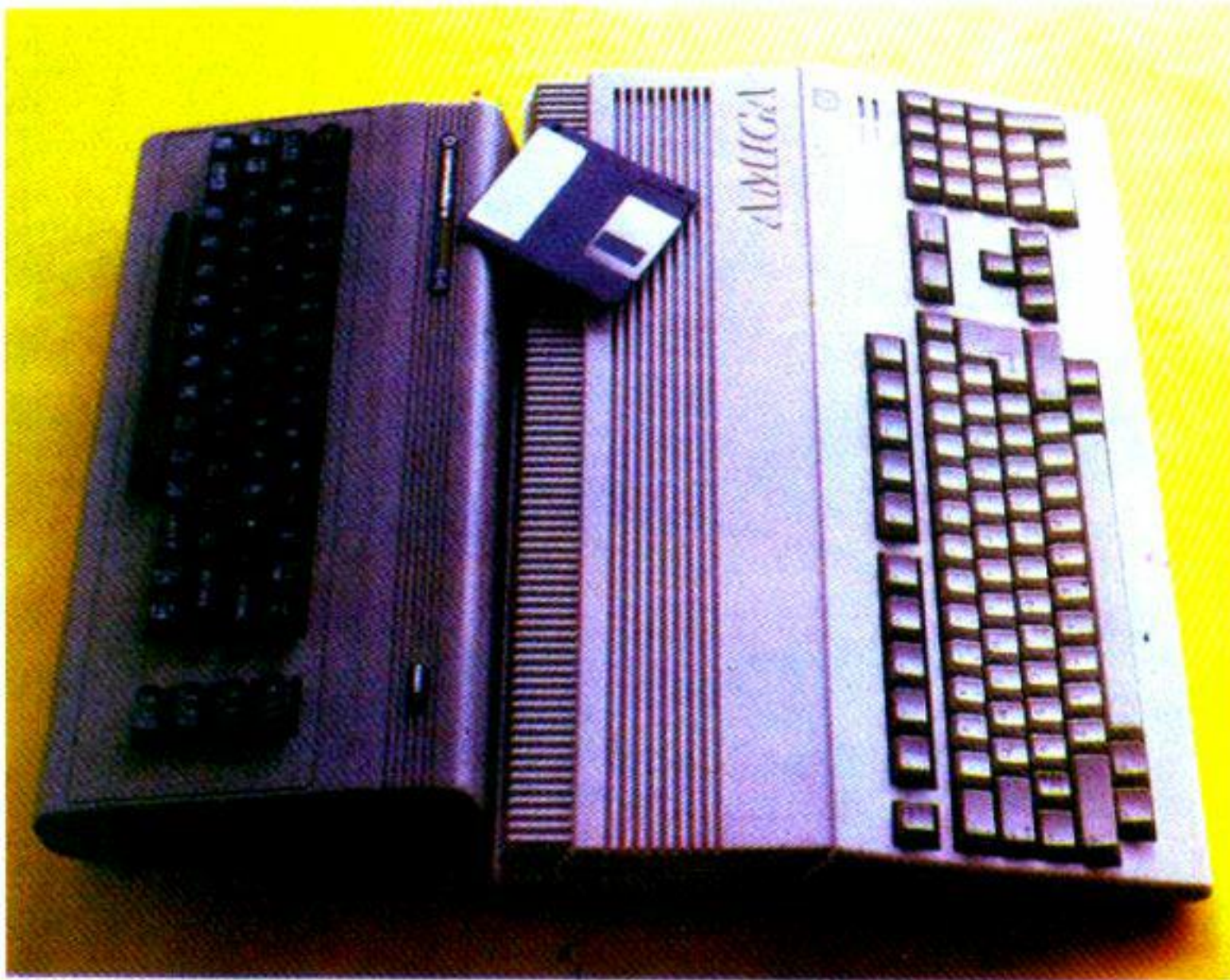
Se, però, dal punto di vista hardware il discorso non faceva una grinza (128K Ram, tre computer incorporati, di cui uno compatibile CP/M, 80 colonne e così via) fu commesso un grave errore trascurando il settore software.

La compatibilità con il C/64, paradossalmente, si dimostrò un vero e proprio cavallo di Troia: le software house, infatti, che devono pensare solo al profitto, decisero, in elevata percentuale, di continuare a produrre, e vendere, software per il C/64 perchè *chi possiede un C/128 possiede "anche" un C/64* ed è quindi interessato allo stesso modo di un sessantaquattrista a procurarsi nuovo software.

La possibilità delle 80 colonne, poi, non veniva adeguatamente sfruttata dall'utente finale che, nella stragrande maggioranza, si limitava ad usare il C/128 nel modo 40 colonne; ma, funzionando in questa modalità, all'utente finale non interessa sapere se un certo programma sta girando in modo 64 oppure in modo 128!

Il C/128, quindi, non rappresentava altro che un C/64 più grande, ma privo del carburante indispensabile per farlo funzionare al massimo: il softawre specifico.

Si capì, finalmente, che nessun computer che, grosso modo, si limitasse a fare le stesse "cose" del C/64 avrebbe avuto successo.

Ed è esaminando il mercato, la concorrenza e, magari, gli errori passati, che nasce l'Amiga.

Perchè Amiga

L'idea di un computer come l'Amiga nasce circa due anni fa, per una serie di circostanze concomitanti: da un lato la concorrenza, benchè non rappresenti un serio pericolo per il target tipico del C/64, dimostra di saper affilare le armi pur se cade spesso e volentieri in errori ben più gravi di quelli commessi (ne siamo sicuri: in buona fede) dalla Commodore.

La Sinclair, proponendo un aborto di tastiera per lo Spectrum e, soprattutto, ritardando la presentazione di stampanti adeguate e di memorie di massa decorose, si brucia la carta del settore professionale e compromette irrimediabilmente l'esito dello stesso QL. Le note vicende finanziarie della casa inglese impediscono, poi, di considerare pericolosi i piccoli computer.

Lo standard MSX, proposto ad un prezzo proporzionalmente inaccettabile, non sembra riscuotere il consenso generale.

Passando a tempi più recenti, inoltre, non può esser considerata una minaccia nemmeno la proposta di "nuovi" computer basati su vecchi microprocessori ad 8 bit che presentano il solo vantaggio di porre in evidenza, ancora una volta, l'inaffondabilità del C/64.

Ma, prima o poi, qualcuno si muoverà ed è bene prendere le opportune precauzioni; anzi, qualcuno si è già mosso, passando dalla Commodore ad un'altra azienda del settore, e non è uno qualunque, ma proprio colui che sovrintende al progetto del successore del C/64.

I provvedimenti devono quindi essere tempestivi: un solo giorno di ritardo può compromettere il futuro dell'azienda. Si riprende il progetto lasciato in sospeso, si esamina ciò che, presumibilmente, potrà esser prodotto altrove dal dirigente dimissionario e si decide, ovviamente, di progettare un computer "migliore": nasce l'Amiga.

In America, come si sa, il fruscio dei dollari fa perdere la testa ed in un turbinio di qualche milione di dollari schiere di tecnici si affannano alla progettazione di Amiga; dapprima un'intera stanza si riempe di schede, cavi, monitor, simulatori, computer dedicati che hanno il compito di simulare le funzioni che il prossimo computer dovrà avere; poi, un po' per volta, alcune grosse schede vengono ridotte alle dimensioni di un chip, altri cavi vengono sostituiti da un più pratico bus su un circuito stampato, alcune procedure software vengono alleggerite, oppure sofisticate, comunque ottimizzate.

Ma il fruscio dei dollari è sempre in agguato; le banche premono, vogliono garanzie, vogliono controllare che le cifre prestate per la realizzazione del progetto possano effettivamente essere restituite.

Uno sforzo sovrumano consente di presentare, commercializzare e rendere credibile l'Amiga 1000; ma, per prudenza, è bene lasciare ancora su disco una parte consistente del software di base (il famoso dischetto di Kickstart). In caso di migliorie sarà molto più semplice rendere disponibile il nuovo dischetto anzichè sostituire la Rom.

Con queste premesse, all'inizio dell'estate dell'86, giunge in Ialia l'Amiga 1000, ed uno dei primi esemplari (con alimentazione a 110 volt, Ram di appena 256 K) desta, qui in Redazione, i primi stupori per la grafica, il suono ed altre amenità alle quali ci saremmo abituati presto.

Resta da spiegare, finalmente, il perchè del nostro riserbo a parlarne diffusamente.

Pur comprendendo le difficoltà sofferte dalla Commodore, relative ad un periodo decisamente non facile (sul quale, ormai, si può addirittura ridere, esorcizzando lo scampato pericolo), non abbiamo mai perso di vista il nostro scopo principale che riteniamo sia quello di fornire una corretta informazione ai nostri lettori: l'anno scorso il fenomeno Amiga consigliava una certa prudenza ed era nostro dovere limitare entusiasmi acritici. Il modello che aspettavamo di "trattare" era, appunto, il 500, decisamente più idoneo, almeno per il mercato italiano, per una rapida e soddisfacente diffusione.

La decisione di commercializzare invece, come primo modello, l'Amiga 1000, ci lasciava perplessi perchè ci era sembrato di capire che la produzione si sarebbe concentrata su un modello più economico (il 500, appunto) che necessitava, però, di altre "rifiniture" prima della sua presentazione ufficiale.

E noi abbiamo aspettato, e non si può dire che tale attesa non sia stata, alla fine, premiata.

Le prime impressioni

L'Amiga 500, come dovrebbe essere arcinoto, ha la forma consueta dei moderni computer domestici: una grossa tastiera contiene, al suo interno, il computer vero e proprio, il connettore per l'espansione di memoria e un disk drive da 3 e 1/2; l'alimentatore, piuttosto voluminoso, non poteva che essere esterno.

E' bene porre subito in evidenza che lo spazio richiesto per la collocazione del sistema è ben maggiore di quello richiesto da un C/64. A parte il mouse, che necessita, da solo, di un'area libera, e facilmente accessibile, di circa 40x40 centimetri, c'è da notare che l'alimentatore è dotato, finalmente, di interruttore; per accendere e spegnere il computer è quindi indispensabile averlo a portata di mano, vale a dire sul tavolo. Il monitor 1081 (a colori, dal suono stereo, in una parola: ottimo) si connette al computer mediante un cavo piuttosto "duro" e difficilmente flessibile che richiede quindi un certo spazio. Tanto per dare un'idea, il 500 che possiedo è posizionato su un tavolo profondo ben 75 centimetri (contro i 60 cm. dei tavoli "medi") e, nonostante ciò, sono stato costretto a realizzare un distanziatore di legno su cui posizionare il monitor e sotto il quale far passare il poderoso cavo. Naturalmente ne ho approfittato per sistemarvi anche l'alimentatore ma è bene tener presente che, per lavorare con l'Amiga, è necessaria un'intera scrivania, specialmente se si decide di procurarsi un secondo drive.

La tastiera è del tipo Qwerty, e presenta minime differenze di posizionamento per ciò che riguarda alcuni segni di punteggiatura; ma, si sa, è questione di abitudine: nonostante, ad esempio, sia presente il comodo ed efficiente tastierino numerico, si tende ad utilizzare la fila alta della tastiera stessa per la digitazione dei caratteri numerici (retaggio del C/64!).

Dieci tasti funzione fanno bella mostra di sè contribuendo a fornire alla macchina quell'aspetto professionale che nulla ha da invidiare ai calcolatori Ms-Dos compatibili (non per nulla l'Amiga sarà compatibile IBM grazie ad una scheda).

Ma ciò che mi preme mettere in evidenza è la totale e assoluta diversità di approccio che un calcolatore della serie Amiga presenta rispetto ad un C/64 al quale siamo tutti noi abituati.

Le differenze concettuali

L'errore principale che si può commettere esaminando un computer come l'Amiga, consiste nel soffermarsi sulla quantità di Ram, sul microporocessore e su altri argomenti che mortificano la concreta innovazione "concettuale" posta alla base dello stesso computer.

Tanto per far paragoni, posso ragionevolmente affermare che un computer Ms-Dos è, per l'utente che lo utilizza singolarmente, soltanto un computer "più grosso" di altri che lo hanno preceduto. Quando, infatti, il sistema Ms-Dos fu presentato, offriva la sola(!) novità di uno standard realmente universale (rispetto al suo predecessore, il CP/M).

Ma tale novità, sotto certi aspetti decisamente rivoluzionaria, poteva non interessare più di tanto una certa categoria di utenti finali che, nel nostro caso specifico di commodoriani, abituati ad una terrificante disponibilità di software per il C/64 (privo totalmente di compatibilità, se non con se stesso) non si sentivano menomati dalla impossibilità di caricare, poniamo, programmi per altri computer dotati, o meno, dello stesso microprocessore.

Insomma, chi ha sperimentato l'Ms-Dos apprezza la maggior quantità di memoria, la maggior velocità di accesso ai dischi ma, tutto sommato, non può che rimpiangere l'intero settore di "intrattenimento" che solo il C/64 può offrire.

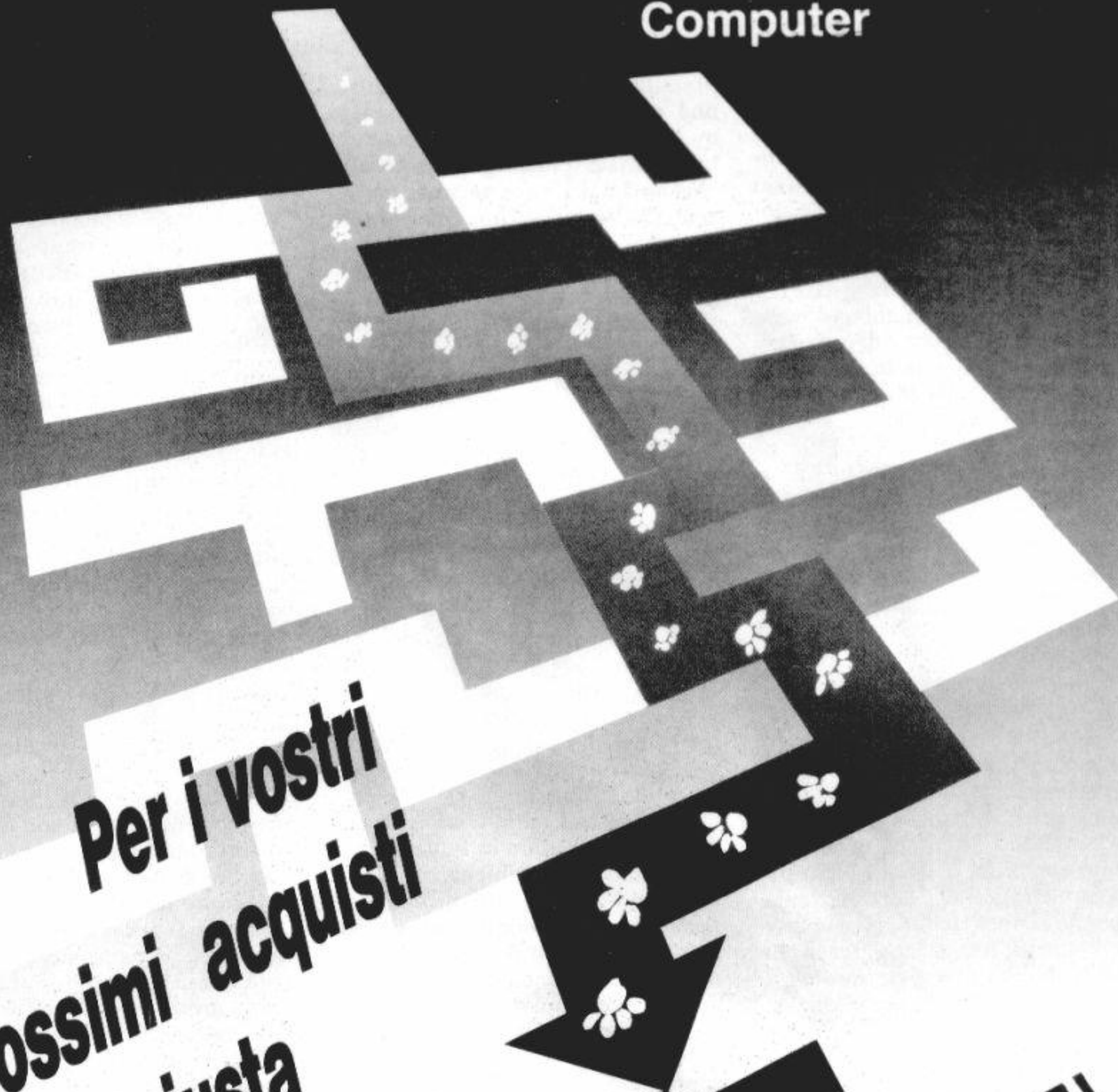
Mouse e tastiera

Amiga "parte" realmente dal punto esatto in cui altri sistemi "arrivano". Abbiamo più volte affermato, sulle pagine di questa stessa rivista, che la scelta di un'opzione porta allo stesso risultato sia che si digiti il numero corrispondente alla scelta stessa, sia che si posizioni la freccia del mouse sul rigo interessato e si prema un pulsante.

Ma pur se il risultato è identico (selezione di un'opzione) la possibilità di utilizzare il mouse, come interfaccia, suggerisce alternative assolutamente impensabili a chi, in precedenza, non abbia mai avuto occasione di usare il "topolino".

L'uso costante del mouse, infatti, apre la strada dell'informatica di massa a categorie di persone che ancora non l'hanno percorsa a causa della presunta difficoltà di approccio.

AVVISO

Per i Signori Negozianti e Operatori Settore

Ma anche per coloro che, con l'informatica, hanno una certa confidenza, la selezione delle scelte tramite mouse non può costituire un aspetto secondario. Chi impara ad usare un mouse si accorge improvvisamente di avere tra le mani uno strumento semplice, ma di incredibile versatilità e, soprattutto, non limitato alla selezione di un menu. In quelle opzioni in cui, per forza di cose, è necessario battere i tasti (registrazione di un file, cambio del nome a un disco e così via), ci si accorge di abbandonare con riluttanza il mouse e di premere i tasti con fastidio: eppure fino a poche ore prima quella era l'unica strada concessa da un C/64!

Chi inizia ad usare l'Amiga scopre con sorpresa che lo stesso esame della directory, la sua sitemazione ottimale, la selezione dei colori e, in una sola espressione, la "preparazione" della macchina, *costituisce, da sola, un autentico divertimento.* Passando, poi, al disegno delle icone per personalizzare i nomi di programmi e di file, il tempo rischia di trascorrere piacevolmente senza fine.

Il mouse, infatti, è un'interfaccia che, in brevissimo tempo, consente di raggiungere, nel campo grafico, una precisione ed accuratezza decisamente superiori a quelle ottenibili con tavolette grafiche del tipo Koala, ben note ai nostri lettori; solo che, riferendomi all'Amiga, non penso che sia possibile proporre hardware aggiuntivo che consenta risultati migliori.

Il simpatico apparecchietto, insomma, non può essere considerato un lezioso accessorio ma, a ben vedere, è una parte integrante del "sistema" che, privo di mouse (e della "filosofia" ad esso legata), perde la sostanziale caratteristica che lo separa da un qulunque Ms-Dos che appare, all'improvviso, arcaico.

E' ben vero che anche per Ms-Dos esiste il cosiddetto ambiente "window" gestibile, anch'esso, da mouse, ma è inevitabile sottolineare che tale ambiente è stato proposto "dopo" la venuta dell'Ms-Dos e la sua gestione risente della struttura che, all'origine, non era destinata a questa specifica applicazione.

Dietro il mouse

Naturalmente, oltre alla gestione del mouse, è stato inevitabile provvedere ad una struttura, hardware e, soprattutto, software, che consentisse una reale innovazione. Ecco, dunque, che in fase progettuale viene subito scartato il microprocessore 8086 che, anche nelle sue versioni più recenti (80286, 80386) non può garantire quella versatilità, quei rapidi spostamenti di blocchi di Ram, quella facilità di spostarsi da un punto all'altro della memoria, la possibilità (reale) di multiprogrammazione. La scelta, inevitabile, cade sul 68000 che ha già dimostrato di possedere un'architettura snella e che già permette di verificare, su altre macchine, le potenzialità di cui dispone. Non a caso il Macintosh lo utilizza da tempo e sembra portare buona fortuna all'Apple. Altri computer grafici, di classe nettamente più elvata, sono concordi nel definire il 68000 come l'indiscutibile microprocessore per determinate applicazioni.

Ma alla Commodore non si fermano qui: per dare maggior respiro al già potente 68000, gli affiancano coprocessori grafici e sonori, non trascurando la possibilità di collegare, allo stesso bus, altre diavolerie che dovessero inventare in futuro.

Il risultato è un ambiente con possibilità grafiche mai viste prima: le finestre che è possibile definire (in numero e tipo), la sintesi sonora, la multiprogrammazione non sono che alcuni degli esempi di ciò che si è realizzato finora, mentre non si riesce ad intuire ciò che il futuro porterà, soprattutto quando il numero di utenti diventerà consistente.

In definitiva con l'Amiga sarà realmente possibile fare "cose" che con il C/64 non è nemmeno lontanamente possibile immaginare; il Geos, tanto per citare uno degli "ambienti" più familiari presso gli utenti del C/64, non è che una pallida idea di un riassunto di un cenno di ciò che si può realizzare con l'Amiga!

Volendo impostare una... proporzione (e chi è un affezionato cliente Commodore sa cosa dico), possiamo affermare che *il C/64 sta all'Amiga come una calcolatrice tascabile sta al C/64 stesso.*

Le dolenti note

Sarebbe sciocco pensare che un computer, come un qualsiasi prodotto della moderna tecnolgia, abbia solo lati positivi: qualche perplessità c'è sempre e, per correttezza, la segnaliamo.

Anzitutto, il drive; moltissimi programmi richiedono due drive per un versatile funzionamento; è quasi obbligatorio procurarsene un altro (da affiancare a quello "incorporato" nel computer). C'è da dire, comunque, che il prezzo di un drive è nettamente contenuto, specie se si ricorre ai drive di altre marche, che sembrano affidabili come quelli originali.

Un'altra nota di demerito riguarda la lentezza di alcuni trasferimenti di dati da e verso il disco. Ci hanno assicurato, però, che è una questione di software e presto sarà disponibile, su disco, una procedura che ottimizzi le operazioni I/O, con particolare riferimento ai drive.

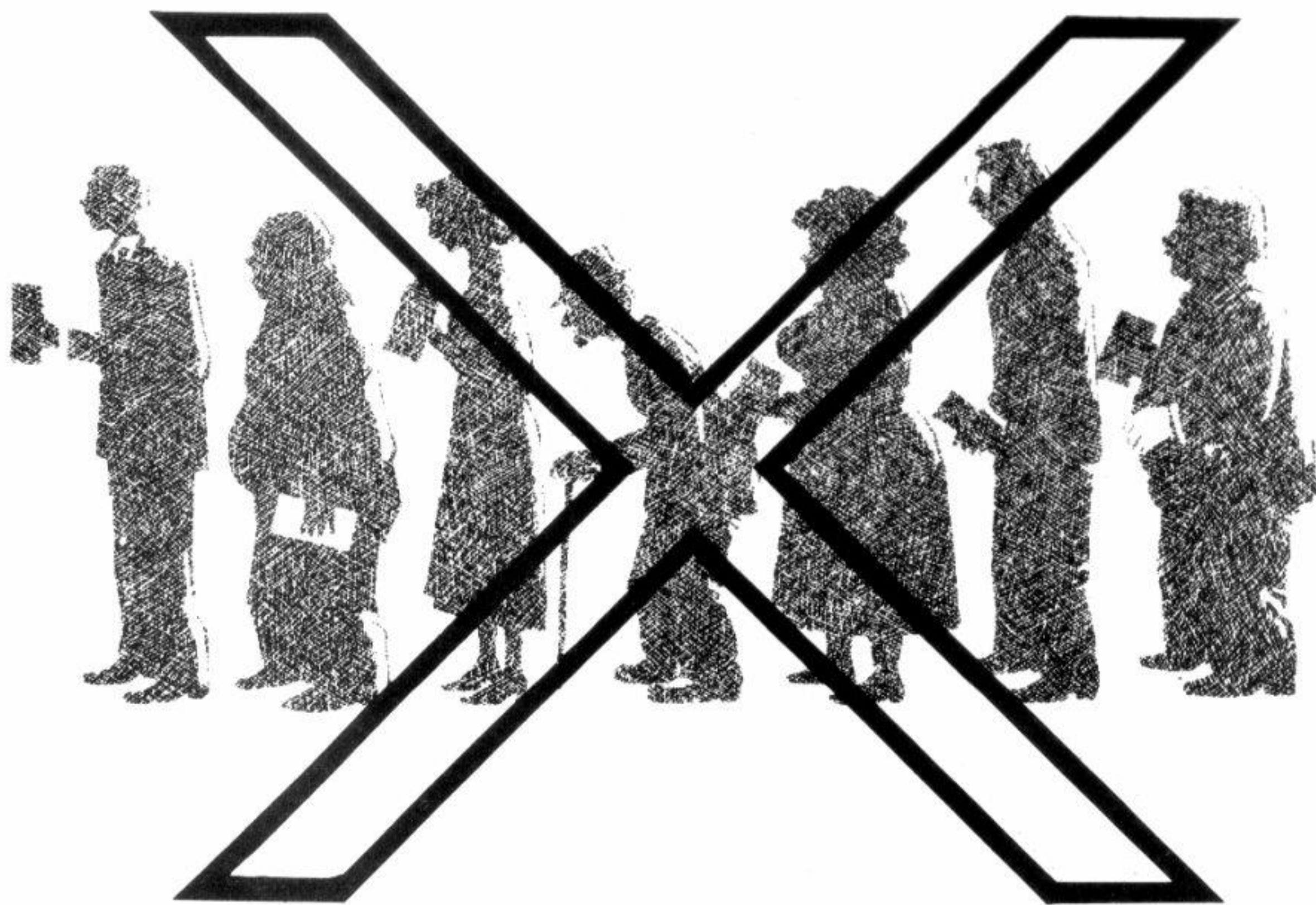
Il monitor dovrebbe essere pressochè indispensabile: si vocifera di un modulatore che consentirebbe l'uso di TV color domestici, ma dovrebbe esser noto che, a certi livelli, la qualità grafica ne soffrirebbe.

In definitiva, non ci si illuda di operare "completamente" con un'Amiga portandosi a casa la sola confezione base. Sarebbe opportuno preventivare, a breve termine, altre spese successive che, nell'ordine, dovrebbero riguardare l'espansione di memoria a un mega, il monitor, ed un secondo drive.

Nonostante tutto, però, il prezzo dell'Amiga 500 (versione base) si aggira sul milione e fa sicuramente venire l'acquolina in bocca a coloro che incominciano a stufarsi del C/64.

Per le stampanti, poi, ci dovrebbe essere presto una grossa novità per un modello laser (avete letto bene: laser) a basso costo che dovrebbe spalancare le porte al Desk Top Publishing, il word processor delle ultime generazioni.

Ma ci hanno pregato di non diffondere ancora le notizie sul suo prezzo. Posso solo dire che questo, comunicatomi in un orecchio, mi ha fatto spalancare la bocca in un enorme Ohh di meraviglia...

DIVENTA CORRENTISTA POSTALE!

il correntista postale con il postagiro
evita il pagamento della tassa di versamento
e la fila allo sportello...

PER PAGARE LA LUCE, IL GAS, IL TELEFONO,
LA TV ED ALTRO BASTA COMPILARE
E SPEDIRE LE BOLLETTE.

Dischi d'autore

Come divertirsi con un programma di utilità fornito con l'Amiga

di Luigi Callegari

E' noto che l'Amiga, computer che deve soprattutto alla grafica il successo che sta riscuotendo, visualizza la Directory di un dischetto in modo anche non convenzionale: oltre al sistema standard (visto sul numero scorso) che consente la visualizzazione dei nomi dei file presenti su disco, è anche possibile associare, a ciascun file, un mini-disegno personalizzato, tracciabile grazie al programma IconEd che ci apprestiamo a descrivere. Questo, infatti, permette di disegnare ideogrammi, od icone che dir si voglia, usate da Intuition.

Come lanciare il programma

Per accedere al programma in oggetto è possibile percorrere due strade:

• ***Da Workbench*** si deve selezionare il cassetto "System" (oppure "Utilities", a seconda della versione che possedete) e poi, nel quadro che compare, "Iconed". Con il termine "selezionare" intendiamo, come abbiamo detto negli articoli precedenti, posizionare la freccetta del mouse sull'icona desiderata e pigiare due volte consecutive il tasto di sinistra dello stesso mouse.

• ***Da CLI (Command Line Interface),*** per i possessori delle versioni aggiornate, è possibile lanciare concorrentemente l'editor di icone con il comando:

```
iconed
```

Per le versioni meno aggiornate, invece, si deve usare:

```
RUN system/iconed
```

non dimenticando, ovviamente, di premere il tasto Return alla fine della riga.

Diciamo subito che la versione del programma qui descritta è la 1.2, ma anche le "vecchie" versioni di Workbench dispongono di programmi praticamente identici, pur se con qualche "bug" in più...

Come usare IconEd

Appena partito, il programma presenta alcune note d'uso (rigorosamente in inglese), specificando che vi sono cinque tipi di icone: disk, drawer, tool, project, garbage.

I rispettivi esempi sono le icone di: disco workbench, icona di directory System, icona di programma Iconed, icona di testo generato con notepad, icona del bidone dell'immondizia (trashcan).

Tali distinzioni vanno ben comprese, perchè non sono puramente accademiche e riguardano sia l'uso pratico di IconEd, sia del sistema AmigaDOS in generale.

Benchè debba esser implicito, è bene precisare che le icone, come qualsiasi "oggetto" presente in un disco, sono formate da byte che, qualunque sia la loro quantità, richiedono l'occupazione di una zona magnetica del disco. Poichè il file e la sua icona di riferimento sono due "cose" diverse, nella directory, richiamata da CLI, figurano appunto due file apparentemente separati e indipendenti tra loro. Anche "sotto" Workbench i due oggetti sono separati (e lo vedremo tra breve) ma, per come sono visualizzati, sembrano essere un unico oggetto.

Le icone del tipo "garbage" (come il bidoncino dell'immondizia ove si ripongono le icone dei programmi o dei file da cancellare), non sono trasportabili in una directory. Ciò significa, ad esempio, che se si tenta di spostare con il mouse il "Trashcan" entro il cassetto "System", il sistema rifiuta di eseguire la consueta procedura.

Le icone di tipo "disk" si riferiscono alla icona relativa ad un dischetto.

Vengono memorizzate nella directory principale del disco con il nome "disk.info". Si noti che quando si formatta un dischetto vergine il sistema automaticamente incide il file "disk. info" che contiene, appunto, i dati necessari alla visualizzazione della icona del disco stesso; memorizza, oltre alla directory "Trashcan" (che contiene i file cancellati dalla directory dei file accessibili, ma non fisicamente eliminati), ed anche il relativo file "Trashcan.info" che contiene i dati per il disegno della pattumiera.

In generale, tutti i file che contengono i dati necessari a workbench per disegnare le icone relative, hanno il proprio nome formato dal nome stesso del file (o del programma al quale si riferiscono) più il suffisso ".info" (abbreviazione di information).

Ad esempio, l'icona di CLI è definita dai dati contenuti nel file denominato "CLI.info".

Questo spiega perchèviene automaticamente aggiunto il suffisso ".info" quando si accede al menu "disk" di Iconed per salvare o caricare file di lavoro per IconEd. Così è

possibile specificare il semplice nome del file o del disco o del programma al quale vogliamo cambiare l'icona.

Si noti che il caricamento ed il salvataggio dei file di lavoro è proprio l'operazione più complicata di IconEd. Una regola generale, consigliata per avere pochi guai, è di specificare i nomi completi dei file/programmi/directory dei quali si vuole elaborare l'icona. E' buona norma non usare troppe abbreviazioni per i nomi (come suggerisce il manuale), perchè oltre alla già nota complessità dei nomi di AmigaDOS, il programma IconED ha alcune (cattive) consuetudini, come ad esempio caricare o salvare solo icone identiche, per tipo, all'ultima caricata dalla directory corrente.

Gli esempi

Vediamo ora qualche esempio. Supponendo di avere il nostro disco "Workbench" (con la tacchetta che consenta la scrittura) inserito nel drive interno (DF0:) useremo, come specificatori di nomi:

DF0:TRASHCAN :per l'icona del bidone della spazzatura.

DF0:SYSTEM :per l'icona a cassettino della directory "system"

DF0:Utilities/calculator :per l'icona della calcolatrice

DF0:SYSTEM/format: per l'icona del programma di formattazione

DF0:DISK :per l'icona del dischetto correntemente nel drive interno.

Per evitare di cancellare accidentalmente il dischetto originale, effettuate una copia su un dischetto vergine, ed operate su questo.

Come si sarà capito, valgono le solite regole per i nomi di AmigaDOS, già esaminate sui numeri scorsi di questa rivista:

• i nomi dei dischetti sono seguiti dal doppio punto;

• tra il nome della directory e quello di un file (o di una sub-directory) deve comparire il segno di diviso (/);

• non vi sono distinzioni tra maiuscole e minuscole, eccetera.

IconEd è in grado di lavorare con un massimo di nove icone contemporaneamente nei nove riquadri visibili. Per indicare l'icona su cui si desidera operare, si posiziona la freccetta e si "clicka" col mouse come al solito. Il disegno selezionato verrà inserito, a sinistra, nel riquadro ingrandito di lavoro, pronto per essere modificato a suon di colpi di bottone del mouse.

Infatti premendo il tasto sinistro, il punto indicato dalla freccetta diviene del colore correntemente selezionato nel menù "Color" (il primo in alto, visualizzabile con la solita procedura: pulsante destro pigiato e freccia sulla scritta in alto "Color").

Il menù "Copy" permette di:

• annullare (Undo = non eseguire) l'ultima operazione compiuta;

• rendere effettiva la modifica dell'icona fatta (Snapshot);

• prelevare da un riquadro (frame) un disegno per lavorarci (From Frame);

• fondere due riquadri tra loro (Merge);

in questi due ultimi casi si specifica il numero d'ordine del riquadro desiderato usando il mouse per evidenziare il numerino interessato, e rilasciando il tasto.

Il menù "Move" permette di:

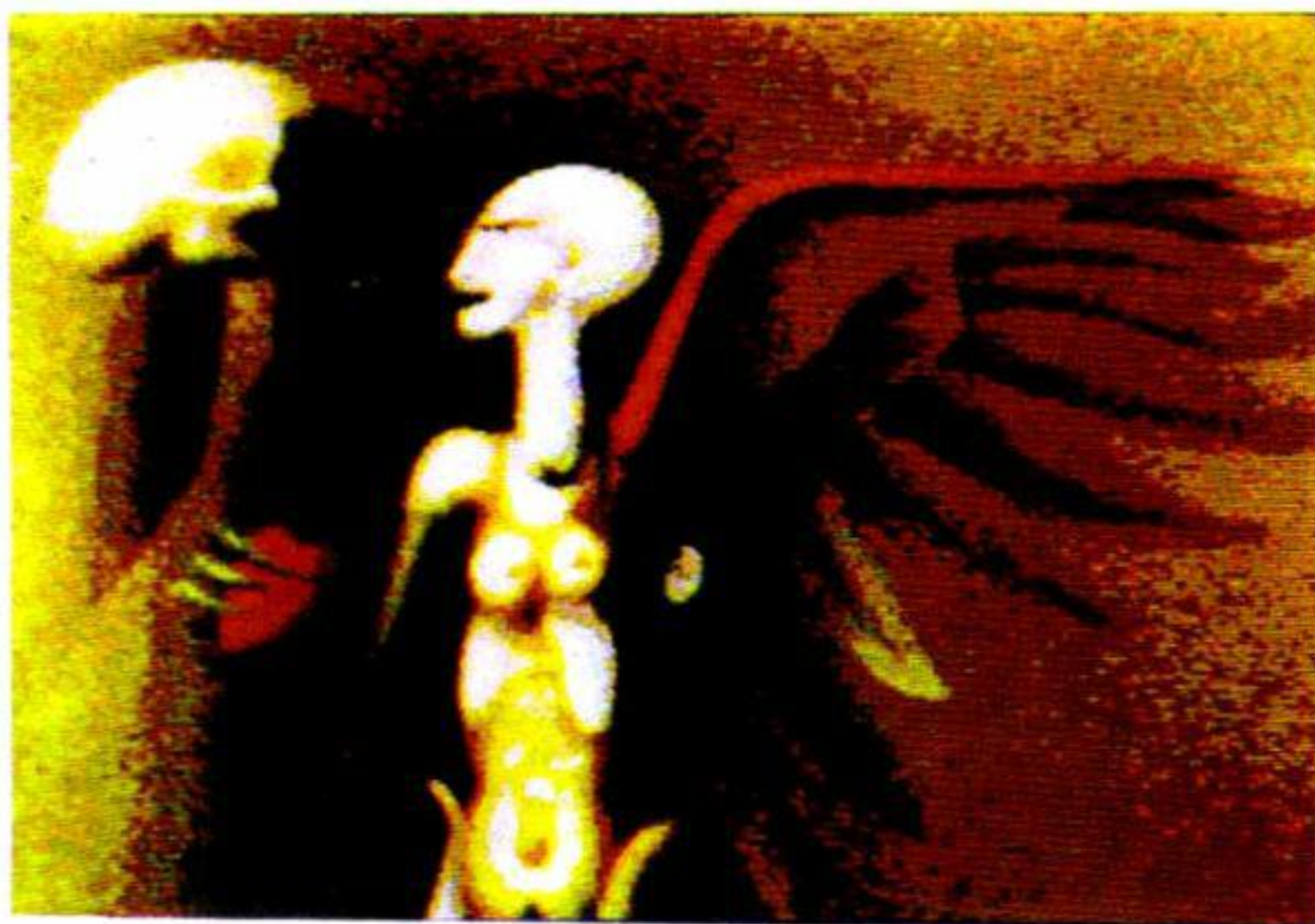
• spostare l'icona nel riquadro;

• di scambiare due riquadri (Exchange) specificando il numero progressivo.

Si noti che la scelta dell'opzione di spostamento fa entrare in un sottomenu che permette di scegliere uno spostamento singolo o continuato ad ogni pressione del tasto (Single - Repeat) l'annullamento (Cancel) e la convalida dell'operazione (OK).

Il menù "Text":

• attiva il modo di scrittura entro il riquadro di lavoro. Inizialmente si de-

ve selezionare col mouse il riquadro ove compare la scritta "text" ed inserire quindi il "testo" che vogliamo fare comparire nell'icona.

Si noti che il "testo" di cui parliamo non sostituisce il nome del file/programma al quale l'icona si riferisce (e che viene stampato automaticamente da Workbench), ma si limita soltanto a inserire eventuali messaggi all'interno dell'icona stessa.

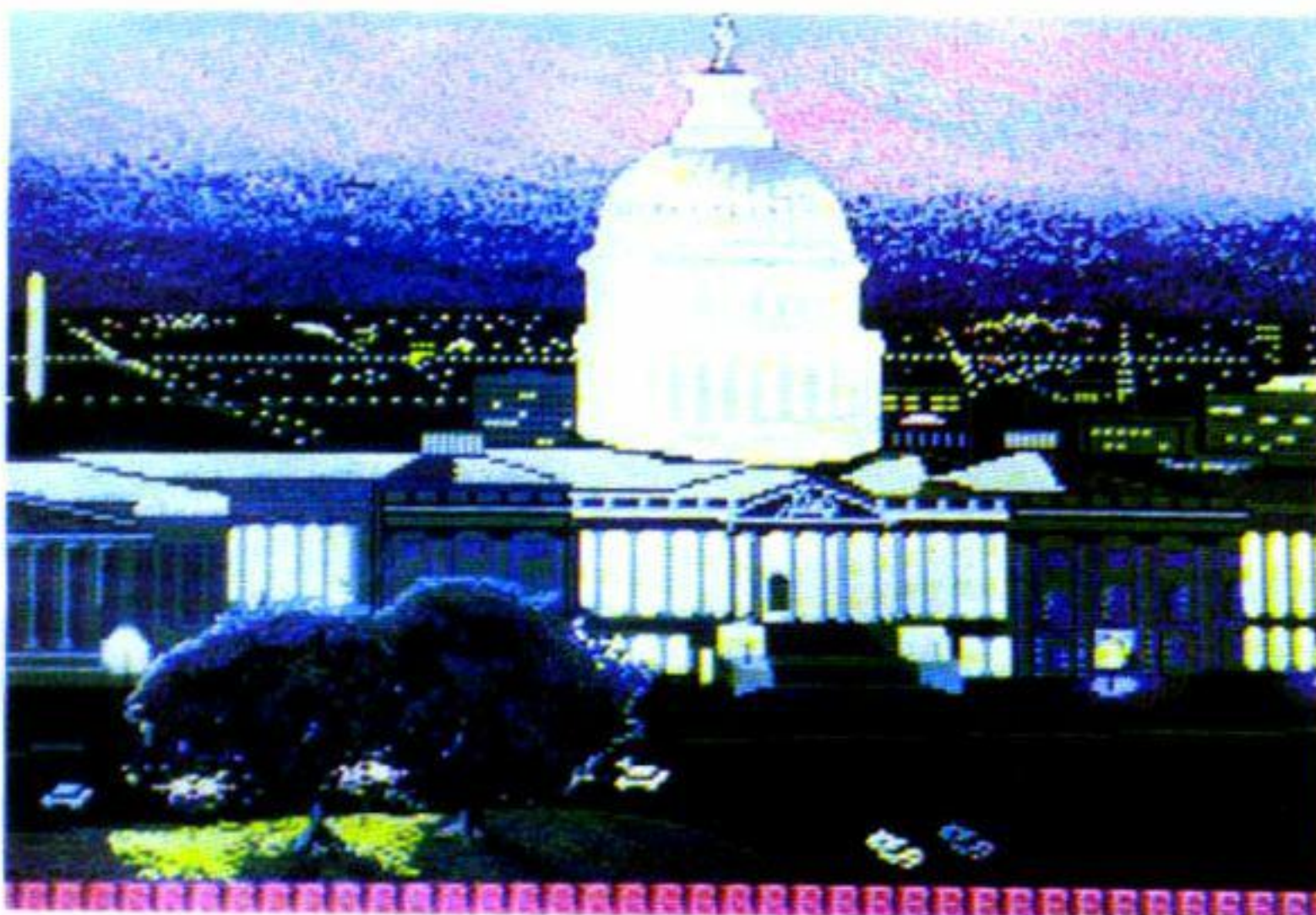
E' possibile, agendo come sempre col mouse, scegliere:

- il colore delle lettere (Foreground);
- il colore dello sfondo (Background);
- la fonte dei caratteri;
- la posizione della scritta (usando un sotto-menu simile all'opzione Move);
- il ripristino delle condizioni iniziali (Reset);
- la cancellazione (Cancel) ed il modo di sovrascrittura:
"Jam1" indica che la scritta usa solo il colore di foreground.
"Jam2" indica che la scritta usa il colore di foreground e di background.
- "Complement" indica che ogni pixel della scritta deve avere il colore invertito rispetto a quello del pixel che sovrascrive (il colore 1 diviene il colore 4 ed il colore 2 il colore 3. In condizioni standard i numeri corrispondono a: blu, bianco, nero, arancione, come da menu "Color".)
- "Inversid" usa il colore di sfondo per le zone non occupate dai caratteri.

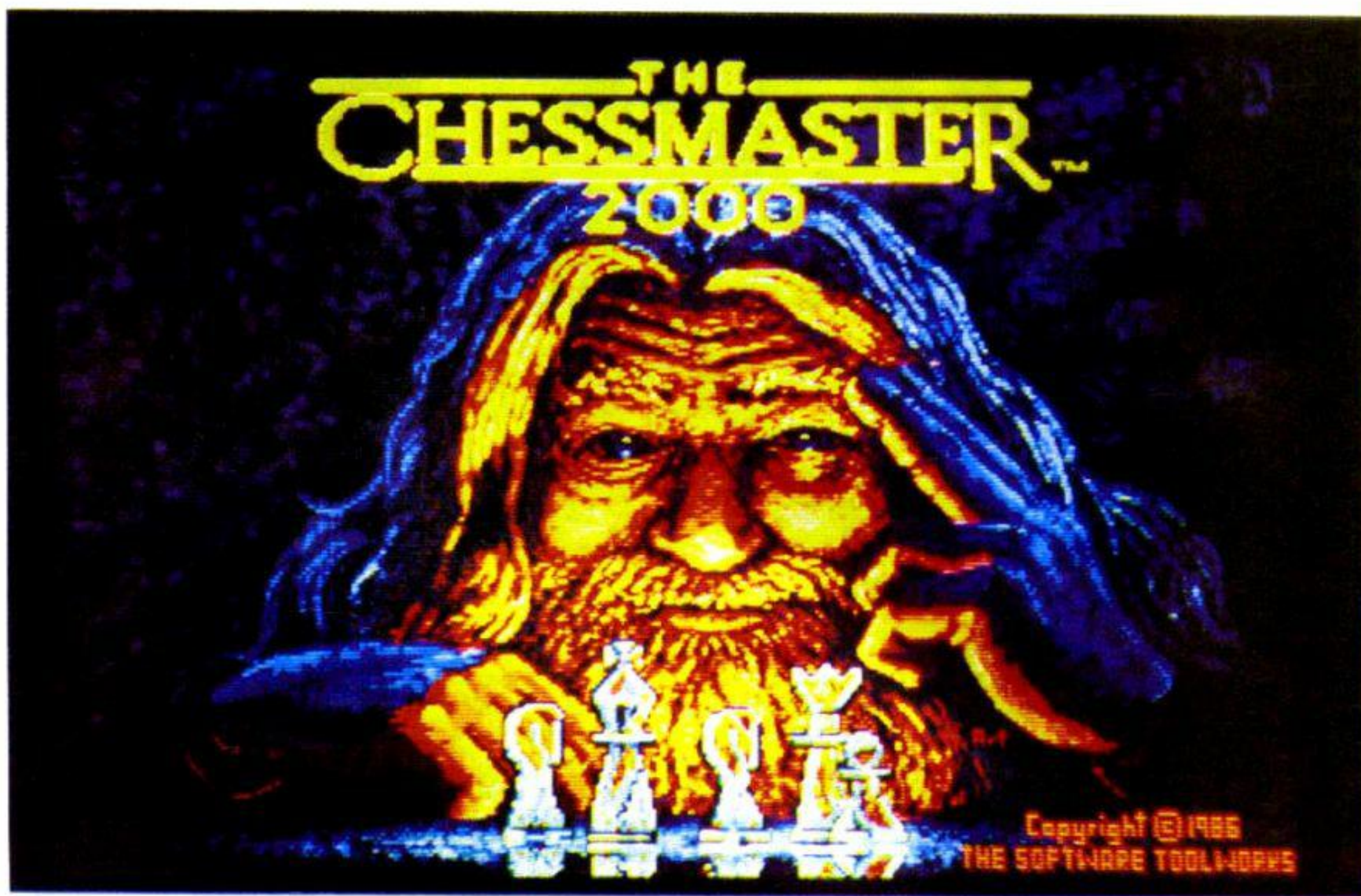
Si noti che per cambiare i colori usati da Workbench si deve usare "Preferences".

Il menù "Disk" permette di caricare e di salvare le icone. Abbiamo già spiegato come si usa. Ci si ricordi che non è possibile modificare con IconED gli ideogrammi associati al mouse (freccia e nuvoletta "Zzzz"), ma la freccia può essere personalizzata da "Preferences".

Il menù "Misc" permette di:
- pulire il riquadro (Clear);
- riempire col colore selezionato una zona (Fill) e di fissare il colore del bordo della icona. Per riempire una zona, dopo avere selezionato l'opzione Fill, si posizioni la freccia nella zona interessata e si prema il pulsante sinistro (si noti che la colorazione interessa zone delimitate da pixel di colori diversi da quello su cui è posizionata la freccetta). Per il bordo, selezionando "1" (tasto destro premuto, freccia sul "Set Bottom Border" e poi su "1", rilasciare pulsante) il sistema lascerà una linea vuota tra l'icona ed il nome associato; selezionando "0", invece, la scritta comparirà "attaccata" al disegnino.

Il menù "Hilite" permette di definire i colori dell'icona quando viene selezionata col mouse. Con "Inverse" i colori vengono invertiti, secondo la legge detta prima (colore 0 con 3 e colore 1 con 2). Con "Backfill" si indica una inversione dei colori per l'icona selezionata come prima, ma le parti dell'icona adiacenti, che sono normalmente visualizzate col colore zero (blu, di regola, come lo sfondo) rimangono dello stesso colore anche dopo la selezione.

Scacco matto con Amiga

Un programma che può dare filo da torcere anche agli scacchisti più esperti

di Michele Maggi

Chiunque sia appassionato di scacchi, e abbia un computer, ha sicuramente uno o più programmi per giocare a scacchi.

Sono in circolazione validissimi software "scacchisti" che raggiungono un elevato livello di gioco senza mai pervenire, tuttavia, ad un livello imbattibile.

Anni fa, durante il mio primo corso di informatica di base, parlando con l'insegnante circa l'abilità dei computer nel campo degli scacchi, mi sentii dire che un computer non sarebbe mai stato un gran giocatore di scacchi a causa di pochi e semplici motivi.

In primo luogo un software "scacchistico" è in grado di elaborare un numero enorme ma FINITO di combinazioni, pur se ad una notevole velocità.

In secondo luogo, una macchina è del tutto priva dell'esperienza, e della fantasia, tipiche di ogni giocatore umano.

Qualsiasi giocatore alle prime armi ha ben poche probabilità di vincita, ma chiunque abbia esperienza in strategie scacchistiche non dovrebbe faticare a battere un computer.

Diverso sarebbe il discorso se il computer avesse la possibilità di archiviare, partita dopo partita, le sequenze delle mosse e gli esiti di ogni partita.

Sarebbe possibile, in questo modo, creare una "banca dati" scacchistica dalla quale il computer potrebbe at-

tingere informazioni ed imparare sempre più.

Un sistema di questo tipo, presupponendo una certa dose di "intelligenza artificiale", potrebbe, in breve tempo, arrivare a livelli di gioco e strategia molto alti.

Per ciò che riguarda i software specifici per Amiga, un posto d'onore spetta senz'altro a Chessmaster, un programma che aggiunge, ad una notevole abilità di gioco, un numero considerevole di opzioni in modo da rendere piacevole ogni seduta di gioco.

La veste grafica, particolarmente curata, è senz'altro la prima cosa che attira l'attenzione; la possibilità di scegliere la scacchiera, a due oppure tre dimensioni, aggiunge un tocco di realismo alla partita stessa, mentre la possibilità di modificare nelle minime sfumature il colore dei pezzi permette di personalizzare la scacchiera in funzione dei propri gusti: la "personalizzazione" della partita, soprattutto le prime volte, è un vero e proprio passatempo, molto divertente.

Le opzioni sono tantissime e permettono numerose varianti sia grafiche che "strategiche"; ecco una breve descrizione di quelle principali:

Dalla schermata di gioco, premendo il bottone destro del mouse, si entra in un "menu mode" che contiene quattro sottomenu:

- *Playing*
- *Board*
- *Display*
- *Extras*

L'opzione Playing consta di una quindicina di scelte relative al gioco vero e proprio, come ad esempio la scelta dell'avversario: è possibile giocare contro il computer, contro un amico oppure assistere ad una partita che il computer gioca contro se stesso.

Fra le "features" più interessanti di questo sottomenu figurano la possibilità di portare indietro la partita di una (o più) mosse, di scegliere il livello di gioco (1-12) e di caricare o registrare una partita.

L'opzione Board serve per gestire la scacchiera in due o tre dimensioni, per sceglierne i colori e per ruotarla.

L'opzione Display consente di ottenere suggerimenti e altre facilitazioni dedicate a chi intende imparare a giocare.

L'opzione Extras consente di analizzare una partita, di stampare le mosse effettuate e di trovare la strategia più breve per uno scacco matto.

Va detto che molte opzioni conducono in sottomenu che a loro volta conducono ad altre possibilità; da ciò si desume facilmente che il numero di opzioni del programma è davvero notevole e perfettamente in linea con la filosofia Amiga.

Un'opzione truffaldina

Come tutti sanno l'Amiga viene sempre più utilizzato anche per motivi professionali in molti uffici; e per chi è costretto, otto ore al giorno, a digitare ed elaborare dati, la tentazione del gioco è forte.

Per questo motivo, per evitare sgradevoli rimproveri da parte del capufficio nel caso si venga colti in flagrante a giocare, esiste una particolare opzione (attivabile premendo il tasto Amiga destro e quello "P") che, a richiesta, simula Amigacalc (un package professionale) con annessi calcoli e risultati; ma, niente paura, appena il capo sarà andato via sarà sufficiente clickare il bottone del mouse per riprendere la partita...

Buon divertimento e... non lavorate troppo!

L'asino di Buridano

Comprare o non comprare un moderno computer a sedici bit?

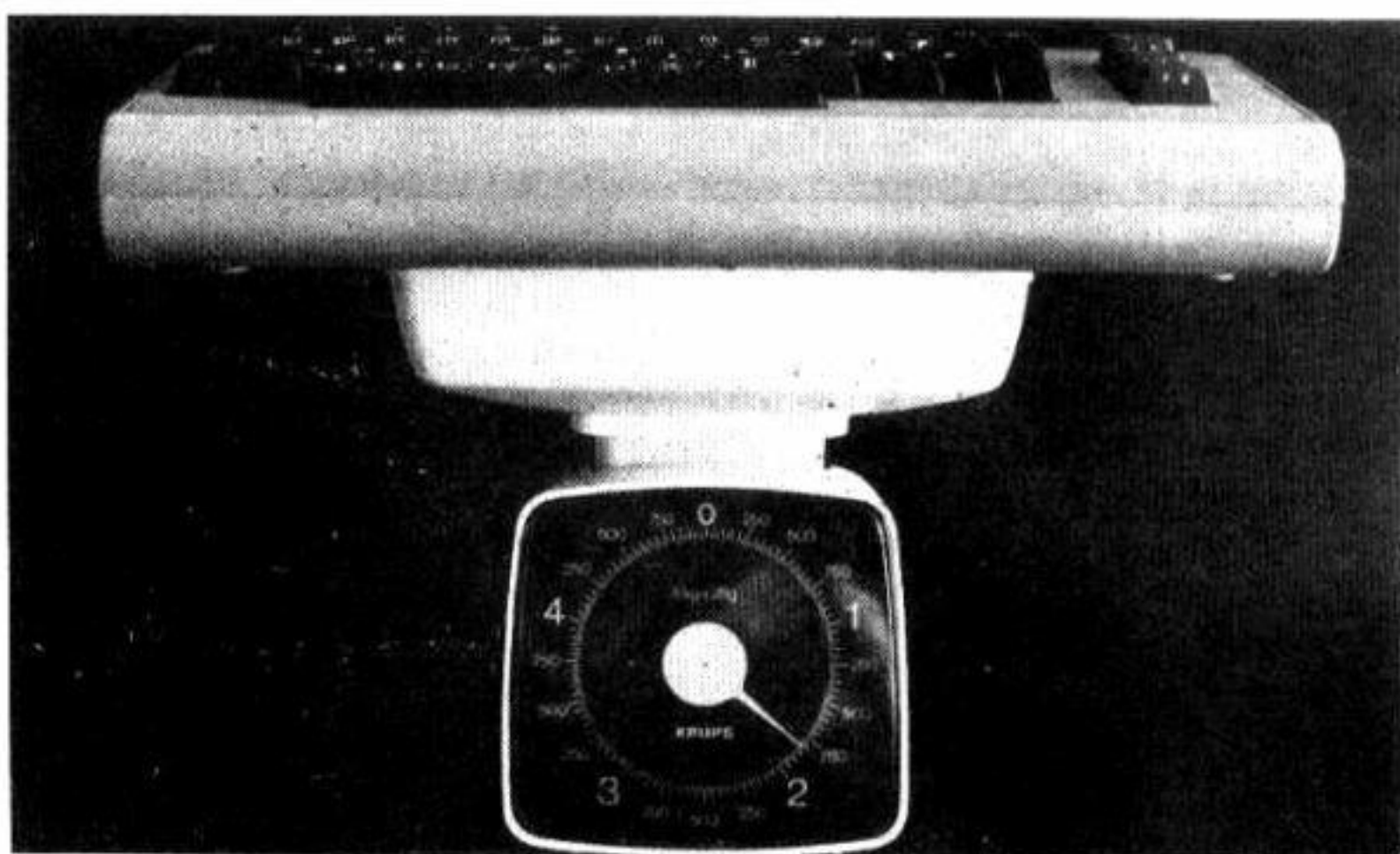
di Paolo Agostini

Il filosofo Giovanni Buridano (ca. 1300-1358), formulatore di una teoria sulla volontà ("voluntas est intellectus et intellectus voluntas"), è rimasto noto per l'apologo dell'asino, che davanti a due mucchi di fieno uguali è incapace di scegliere e muore di fame.

Anche gli hobbisti evoluti sentono che i grigi 8 bit del C-64 cominciano ad essere superati dal continuo apparire di nuovi modelli, sempre più evoluti nello splendore dei 16 bit e con una notevole disponibilità di memoria RAM, per cui ci si trova un po' nella stessa situazione dell'asino. Anzi, per dirla con Dante: "Intra due cibi, distanti e moventi d'un modo, prima si morria di fame che liber'uomo l'un recasse ai denti" (Paradiso IV, 1-3).

Che cosa c'è dunque al di là del C-64? Quale sarà il nostro prossimo computer, al quale dedicheremo ore, giorni e notti insonni?

Un po' di storia

Lasciamo che sia la ragione a guidare la nostra scelta. Uno dei criteri sarà quello dell'esame del "cervello" che il computer impiega per svolgere i suoi compiti. I primi processori impiegati potevano operare su "parole" di soli 4 bit, come il 4040 fabbricato dalla Intel. Questi processori erano adatti per semplici applicazioni di controllo e giochi elettronici poco sofisticati; con l'introduzione di processori ad 8 bit iniziò la diffusione dei minicomputer. I processori più comuni per applicazioni domestiche sono stati (e lo saranno ancora, per qualche tempo almeno) lo Z80

(Spectrum), il 6502 (Apple, VIC-20) e il 6510 (Commodore 64).

Con il continuo affinamento delle tecnologie di produzione si sono resi via via disponibili processori sempre più potenti a prezzi sempre più bassi, il che ha visto l'introduzione sul mercato di chip a 16 bit, come l'8086/8088

Z-80 (8 bit, prodotto da Zilog)

Memoria indirizzabile direttamente: 64 kbyte
Istruzione più veloce: 1 microsec (clock a 4 MHz)
Registri generali: sette a 8 bit oltre al set duplicato
Altri registri: 2 registri indice a 16 bit: 2 stack-pointer a 16 bit
Livelli di interrupt: 2

Note: I registri a 8 bit possono essere accoppiati e usati come registri a 16 bit. Il set di istruzioni supporta un'aritmetica a 16 bit.

80286 (16/24 bit, prodotto da Intel)
Memoria indirizzabile direttamente: 16 Mbyte
Istruzione più veloce: 0.2 microsec (clock a 10 MHz)

8086/8088 (16 bit, prodotto da Intel)
Memoria indirizzabile direttamente: 1 Megabyte
Istruzione più veloce: 0.4 microsec (clock a 5 MHz)
Registri generali: 4 a 16 bit
Altri registri: registro base, stack/indice
Livelli di interrupt: 2

Note: Il microprocessore ha 24 modi di indirizzamento degli operandi e può eseguire moltiplicazioni e divisioni a 16 byte signed e unsigned, ha istruzioni di loop e può accedere alla memoria in modo read-modify-write indivisibile. Il processore 8088 ha un bus esterno a 8 bit anzichè a 16 bit e perciò può essere usato in associazione ad altri componenti a 8 bit.

(IBM PC XT e IBM compatibili), a più bit, come lo 80286 a 16/24 bit (IBM PC AT e compatibili) e il 68000. Quest'ultimo è il primo microprocessore con una architettura e un set di istruzioni simili a quelli di un mainframe (maxicomputer): è in grado di indirizzare direttamente un grande spazio di memoria e di trattare dati di 8, 16 e 32 bit. Un "fratello minore" di questo processore viene impiegato sul QL, mentre il 68000 è utilizzato per l'Atari ST, l'Amiga e il Macintosh Apple. Esaminiamo alcune caratteristiche di questi processori:

6502/6510 (8 bit, prodotto da MOS Technology)

Memoria indirizzabile direttamente: 64 kbyte
Istruzione più veloce: 0.5 microsec (clock a 4 MHz)
Registro generale: 1 (accumulatore) a 8 bit
Altri registri: 2 registri indice (X/Y) a 8 bit; 1 stack-pointer a 8 bit
Livelli di interrupt: 2

Note: Manca di istruzioni per la manipolazione diretta di dati più lunghi di 8 bit, e si supplisce a tale carenza mediante l'impiego della pagina più bassa della memoria (pagina zero) accoppiandone i byte per l'indirizzamento a 16 bit.

Registri generali: 8 a 16 bit
Livelli di interrupt: 2

Note: L'80286 è compatibile con i meno potenti 8086/8088 e può eseguire i programmi scritti per macchine che impieghino questi processori con poche o nessuna variazione (si tratta della famosa compatibilità dei programmi IBM che possono essere fatti girare sia sul modello XT che sul modello AT). La differenza principale consiste nella velocità di esecuzione e nel fatto che l'80286 può supportare sistemi multiutente.

68000 (16/32 bit, prodotto da Motorola)

Memoria indirizzabile direttamente: 16 Megabyte
Istruzione più veloce: 0.5 microsec (clock a 8 MHz)
Registri generali: 16 da 32 bit
Altri registri: registro stack utente
Livelli di interrupt: 7

Note: è in grado di trattare dati di 8, 16 e 32 bit, ha 16 registri di 32 bit ognuno, alcune istruzioni atte a facilitare la compilazione di linguaggi ad alto livello. In modo supervisore può essere usato per evitare che programmi non privilegiati accedano a certe aree di memoria oppure diano inizio direttamente ad operazioni di Input/Output. Ha inoltre la possibilità di lavorare in "multitasking", e prevede una serie di utility per la sincronizzazione in ambiente multiprocessore.

Prime considerazioni

Questi pochi parametri sono certamente insufficienti per fornire un'idea precisa sulle possibilità delle macchine basate su tali processori; se non altro forniscono le prime pietre di paragone, vale a dire la velocità operativa e la quantità di memoria indirizzabile direttamente. Maggiore è tale quantità di memoria e maggiore è anche la capacità della macchina di trattare un grande numero di dati. Una macchina in grado di indirizzare direttamente 16 Mbyte può gestire array di grandi dimensioni contenendoli tutti in memoria. Inoltre il costo sempre più ridotto dei chip di memoria fà si che il possesso di macchine in grado di essere espanse divenga sempre più appetibile per gli hobbisti evoluti, soprattutto per ciò che riguarda la gestione della grafica in alta risoluzione (specie se a colori) e la memorizzazione di lunghe frasi sintetizzate.

L'altro criterio di esame è il rapporto prestazioni/prezzo. Spesso il costo del computer non è dovuto alla prestazioni che offre, ma a fattori diversi: nome, prestigio, politiche di mercato, servizio postvendita, e così via. Nel biennio 1985/86 c'è stato il boom dello standard MS-DOS, e molti si sono affrettati ad acquistare l'IBM XT. Il costo elevato dei computer originali IBM ha però spinto molti all'acquisto dei cosiddetti IBM-compatibili: come in tutte le cose c'è chi ha avuto fortuna e chi no. Spesso i compatibili, prodotti in Estremo Oriente, lasciano a desiderare per ciò che riguarda la qualità dei componenti. Inoltre alcuni fabbricanti di Hong Kong e Taiwan impiegano componenti irreperibili in Italia. Accade che talvolta, a causa dell'imprevidenza di importatori poco scrupolosi (che non prevedono un servizio post-vendita), sia impossibile far riparare il prezioso gioiellino pagato, è vero, una cifra modesta rispetto al costo dell'originale, ma che "cifra" rimane.

Passata, però, la prima euforia, ci si è accorti che il compatibile XT, in fondo, non era quel gran salto qualitativo ("upgrade") che ci si sarebbe aspettato, forse perchè eravamo ormai "viziati" dall'ottima qualità raggiunta dai programmi circolanti sul C-64, oppure perchè sul mercato erano frattanto apparsi modelli più avanzati dell'ormai "anzianotto" XT. Oggi gli hobbisti più evoluti hanno cominciato a guardare sospirosi gli AT (dove il prezzo, nel caso di un modello "compatibile" dell'Estremo Oriente, inizia dai quattro milioni e va crescendo a seconda degli "optionals" desiderati). Effettivamente l'AT rappresenta un punto di arrivo particolarmente appetibile, ma anch'esso viene considerato un po' superato essendo già stato posto in vendita un nuovo chip, l'80386, destinato, nelle intenzioni dei produttori, a sostituire il "lento" (!) processore 80286. E' chiaro comunque che questo tipo di macchine non è alla portata di tutte le tasche.

Le alternative

Ma esiste un'alternativa valida? L'apparire sul mercato del processore 68000, che, oltre ad avere un costo contenuto, presenta una serie di vantaggi non indifferenti, ha mutato il quadro a tutto favore di noi hobbisti. I produttori di hardware, fiutato l'affare, si sono precipitati a capofitto nell'impresa di conquistare quelle fette di mercato che la presumibile prossima scomparsa di Sua Maestà il Commodore 64 avrebbe reso disponibile.

I primi sono stati l'Apple MacIntosh, che in Italia ha trovato relativamente pochi "aficionados". Al grido di "le roi est mort, vive le roi!" è quindi venuto l'Atari ST, anche questo con poco seguito, ed è ora il turno dell'Amiga Commodore per il quale mancano ancora dati di vendita attendibili. Sembra comunque che (complice l'assoluta mancanza in Italia di una cultura informatica diffusa) la relativa complessità di questi apparecchi abbia il potere di spaventare i potenziali compratori.

Non mancano certamente computer di altissima qualità che utilizzano il processore 68000, ma il loro costo è molto al di fuori della portata delle nostre tasche: valga per tutti la menzione del superblasonato STRIDE 460 con clock a 12 MHz, 12 MByte di memoria e un Winchester da 448 MByte (no, non è un errore tipografico, sono proprio quattrocentoquarantotto mega!), il cui prezzo si aggira sui duecento milioni.

Crediamo sia utile, per una scelta oculata, mettere a confronto, con una tabella i dati tecnici dei computer a 16 bit menzionati. Ciò non esclude, naturalmente, che esistano sul mercato anche altri computer, parimenti validi, anche se meno diffusi di quelli indicati. La scelta finale la farà l'utente e, come ben sappiamo, spesso non è la ragione ma il cuore a guidare la scelta...

Elemento	IBM XT	IBM AT	ATARI ST	SINCLAIR QL	APPLE McINTOSH	COMMODORE AMIGA
Processore	8086/8088	80286	68000	68008	68000	68000
Memoria ROM			192 Kbyte	48 KByte	64 Kbyte	192 Kbyte
RAM interna	256 Kbyte	512 Kbyte	ST520: 512 KB ST1040: 1 MB	128 Kbyte	128/512 Kbyte	256 Kbyte
Espandibilita' RAM interna	640 Kbyte	1 (16) Mbyte	no	no	1/2/4 Mbyte	512 Kbyte
Esp. RAM este.	no	no	4 Mbyte	no	no	8 Mbyte
Frequenza del clock	4,77 MHz	6-8 MHz	8 MHz	8 MHz	7,8336 MHz	7,159 MHz
Definizione punti schermo: --monocromat. --a colori	 scheda agg. scheda agg.	 scheda agg. scheda agg.	 640 x 400 320x200/16col.	 no 256x256/8 col. 512x256/4 col.	 512 x 342 no	 640 x 400 320x400/32 col
N. max colori			512 colori	8 colori	bianco e nero	4096 colori
Sprites	no	no	no	no	no	8 sprites
Suono	beeper	beeper	3 voci	beeper	4 voci	4 voci stereo (sintesi voc.)
Segnale per il monitor			RGB analogico	RGB		RGB analogico RGB digitale NTSC composito connettore TV
Interfaccie con l'esterno	Porta seriale Porta parall.	Porta seriale Porta parall.	Centronics RS-232 Disk Drive Hard Disk MIDI ROM-Cartridge Mouse/Joystick	2 x RS-232 Network Joystick Bus di sistema	RS-232 RS-442A Mouse	Centronics RS-232 Drive esterno Hard disk Mouse/Joystick Bus di sistema Conn. stereo
Dischi usati	5 1/4 pollici	5 1/4 pollici	3 1/2 pollici	microdrive	3 1/2 pollici	3 1/2 pollici
Byte per disco	360 Kbyte	1,2 MByte	360 Kbyte	100 Kbyte	400 Kbyte	880 Kbyte
Multitasking	no	no	no	si	no	si
Tastiera	separata	separata	integrata	integrata	separata	separata
No. di tasti	ca. 85	ca. 85	94 tasti	65 tasti	59 tasti	89 tasti

Un'estate gialla

Per i patiti del "Giallo elettronico" una nuovo pieno di avventure per una lunga estate

Computer fiction, electronic novel, interactive novel. Comunque la si chiami, una nuova forma di arte è nata ed attende di essere battezzata.

Così David Lehman dalla rubrica di critica letteraria "Back of the book" dell'autorevole settimanale Newsweek.

Il dibattito sul romanzo interattivo esce ora dalle riviste specializzate per approdare sui settimanali a larga diffusione, almeno negli U.S.A.

In Italia l'uscita de I Gialli Commodore, l'unica collana di interactive novels italiana, è stata accolta favorevolmente dai maggiori organi di stampa, Il Corriere della Sera, La Stampa, La Repubblica, Il Manifesto, come dimostrano gli estratti pubblicati.

Incoraggiati dall'interesse riscontrato, vi presentiamo ora il terzo numero della collana.

In esso troverete:

Il Mistero di Zambesi Waters - Una interactive novel gialla di ambientazione africana.

Si tratta della prima iniziativa italiana rivolta verso questa nuova forma «letteraria» e l'interesse che ha suscitato fa sperare che editore ed autori diano continuità all'iniziativa.

Gli autori, Sandro Certi e Franco Toldi, si occupano da tempo su alcune riviste di *adventures games*, gli stessi autori hanno curato molte rubriche sull'intelligenza artificiale.

il manifesto/giovedì 16 aprile 1987

Le indagini di Thomas Bradly:

- I due cugini
- Un furto d'arte
- Il testimone

Tre giochi di investigazione ambientati nella mitica Chinatown dove nei panni di questo investigatore fuori dagli schemi dovrete misurarvi con pericoli e misteri di ogni genere.

Il Mistero di Zambesi Waters

L'inverno sembrava non volesse finire mai. Avevi trascorso le vacanze natalizie lavorando freneticamente al caso Lange, senza grossi risultati. Il cliente ti aveva poi sospeso l'incarico. La faccenda ti aveva lasciato stanco e insoddisfatto. Avevi bisogno di staccare, ma ogni volta rimandavi la decisione. Quella mattina avevi scorso con disinteresse la posta. Una busta dal francobollo variopinto aveva attirato la tua attenzione...

Sarà seguendo l'invito contenuto in quella lettera che, nel ruolo di Sam Chase, lo stanco detective americano protagonista di questo mistery, atterrerete nell'assolata Bulawayo. Ad attendervi non troverete tuttavia solo chi vi ha invitato.

Presto, a costo della vostra vita, dovrete battervi contro un avversario deciso a tutto pur di impedire che dal passato riemergano terribili fantasmi. Solo i vostri sforzi potranno far fallire i sui progetti. La verità scioglierà così l'oscuro incantesimo che regna ancora sulle vecchie fattorie del nuovo Stato.

Questo genere di videogiochi quasi sempre ispirati da classici della letteratura gialla si è conquistato un grande numero di appassionati ed ora anche in Italia vengono prodotti i primi giochi.

Franco Toldi e Sergio Certi, due insegnanti milanesi con la passione del giallo e l'interesse per l'informatica, hanno realizzato la collana *I gialli Commodore* (in questi giorni è uscito in edicola il secondo numero) e sperano di poter realizzare la versione computerizzata dei *Gialli Mondadori*.

Ogni storia ha una struttura ramificata piuttosto complessa in cui il giocatore, in base alle sue scelte ed agli indizi trovati, segue un percorso ipotetico che lo conduce alla risoluzione dell'enigma. "Per risolvere i nostri gialli — aggiunge Sergio Certi — bisogna avere intuito e intelligenza nell'interpretare le indicazioni che vengono dal testo che descrive le varie situazioni. Il giocatore deve fare molta attenzione nelle sue decisioni perché un particolare ignorato oppure una mossa sbagliata potrebbero allontanarlo definitivamente dalla soluzione.

la Repubblica

Arrivano i Commodoctors!

Una coccarda blu, bianca e rossa, come quella della rivoluzione francese, cambierà radicalmente il modo di garantire l'incolumità dei Computer commodore

La novità grafica si accompagna ad una radicale trasformazione della struttura di assistenza della Commodore Italiana, voluta dal nuovo management della società.

Ecco a grandi linee le caratteristiche delle nuove garanzie commodore: centri assistenza altamente qualificati, che svolgono solo l'attività di assistenza, e non di vendita.

Tempi di riparazione molto brevi; l'utente non dovrà più attendere mesi per le riparazioni del proprio computer.

E chi ha già acquistato il calcolatore con la vecchia garanzia?

Niente paura, la Commodore ha pensato anche a loro, infatti chiunque potrà usufruire dei centri assistenza Commodore, alle condizioni viste prima.

L'iniziativa è stata presa nell'ambito di una politica volta a favorire il rapporto tra la Commodore ed i suoi clienti, in modo da evitare che, una volta acquistato il calcolatore, l'utente, in caso di necessità, abbia difficoltà nel reperire pezzi di ricambio, o debba rivolgersi ad altri per farlo riparare.

Inoltre, presso i centri autorizzati, verrà garantito, oltre la celerità, anche un costo contenuto del servizio di assistenza.

La coccarda, non è quindi solo il simbolo della nuova garanzia, ma rappresenta la rivluzione delle strategie imprenditoriali della Commodore Italiana.

ABRUZZO

Audio Computer
Via Umbria,7
Pescara
Tel. 085/293375

CALABRIA

Service Center
Via Parisio, 25
Cosenza
Tel. 0984/757441

CAMPANIA

Electrical Engineer
Via Supportico Lopez, 5/A
Napoli
Tel. 081/293408

EMILIA ROMAGNA

Centro Riparatori
Via Lenin, 48/I
Carpi
Tel. 059/640770

Centro Riparatori
Via Galvani, 4
Modena
Tel. 059/216602

Dr. Sax
Via D. Creti, 77/C
Bologna
Tel. 051/352539

S.I.R.A. S.r.l.
Via Aniene, 42
Ravenna
Tel. 0544/64223

Maser
Via collegio di Spagna, 10
Ozzano Emilia (BO)
Tel. 051/798448

FRIULI VENEZIA GIULIA

Audio Video Service
Via Gemelli
Pordenone
Tel. 0434/961104

ET Italia
Via Tavagnacco, 89
Udine Tel. 0432/481339

Hitech
Via Nordio, 9
Trieste
Tel. 040/741189

LAZIO

Computer Service Italia
Via Sebino, 49
Roma
Tel. 06/850386

Tecnicomp
Via dei Georgofili, 65
Roma EUR
Tel. 06/5133739

Commodore

TAGLIANDO DI ACCETTAZIONE

PARTE DA STACCARE E INVIARE

Entro 3 giorni dalla data di acquisto

N. 70001552 MOD. C 64

Data limite validità garanzia 31/12/88

Timbro e dati rivenditore

Data acquisto

Informazioni anagrafiche del cliente

Cognome e nome

Indirizzo

Città CAP

Settore merceologico

La mancanza del tagliando sottostante certifica l'avvenuto collaudo a campione dell'apparecchiatura

N. 70001552 MOD. C 64

Timbro e dati rivenditore

RISERVATO AL RIVENDITORE

Commodore

CERTIFICATO DI GARANZIA

PARTE DA CONSERVARE

Anche dopo la data di scadenza

N. 70001552 MOD. C 64

Data limite validità garanzia 31/12/88

Data acquisto

Timbro e dati del rivenditore

Firma rivenditore

IMPORTANTE

Il presente certificato attesta la regolare importazione del prodotto da parte della Commodore Italiana S.p.A. ed è l'unico documento che permette l'utilizzazione dei servizi Commodore ai clienti.

TAGLIANDO DI RIPARAZIONE N. 1

Da compilare a cura del riparatore N.

Mod. C 64

Data limite 31/12/88

70001552

Cod. riparatore

Commodore

TAGLIANDO DI RIPARAZIONE N. 2

Da compilare a cura del riparatore N.

Mod. C 64

Data limite 31/12/88

70001552

Cod. riparatore

Commodore

TAGLIANDO DI RIPARAZIONE N. 3

Da compilare a cura del riparatore N.

Mod. C 64

Data limite 31/12/88

70001552

Cod. riparatore

Commodore

LIGURIA

Siragusa
Via Milano, 41
Genova
tel. 010/261655

LOMBARDIA

Abiservice
Via Ponale, 48
Milano

Catme
Via Severoli, 9
Milano
Tel. 02/4152962

Computer Service
Via Genala, 19/A
Cremona
Tel. 0372/435861

MARCHE

Car
Via Bruno Buozzi, 18
Ancona
Tel. 071/804488

PIEMONTE

Grun A.R.
Via De Sanctis, 126/F
Torino
Tel. 011/2202666

PUGLIA

Viga
Via Domenico Morea, 42
Bari
Tel. 080/413766

SARDEGNA

General Progetti
Centro Commerciale Rejna
Viale Monastir, 110
Cagliari
Tel. 070/283291

SICILIA

Cat Elettronica
Via Ravenna, 7/A
Catania
Tel. 095/438670

CO.AS. Informatica
Via Raffaele Mondini, 3
Palermo
Tel. 091/295209

TOSCANA

G.L.G. Elettronica
Via Pietrasantina, 61
Pisa
Tel. 050/562035

Paolo Paolieri
Via Perfetti Ricasoli, 70
Firenze
Tel. 055/4361720

TRENTINO ALTO ADIGE

Elecomp
Via Druso, 52/A
Bolzano
Tel. 0471/42128

UMBRIA

H.C.H.
Via Ruggero D'Andreotto, 49
Perugia
Tel. 075/22042

VENETO

Capanese Elettronica
Telecomunicazioni
Strada VII Martiri, 101
Padova
Tel. 049/624160

Sistel
Via Decorati
al Valore Civile, 67
Mestre
Tel. 041/935332

Pesente Giovanni
Via Pitagora, 6
Verona
Tel. 045/565988

Il C-64 si affaccia sul mondo

Nella meravigliosa cornice della penisola sorrentina è stato presentato ufficialmente il recente accordo tra Commodore, Seat e Sip

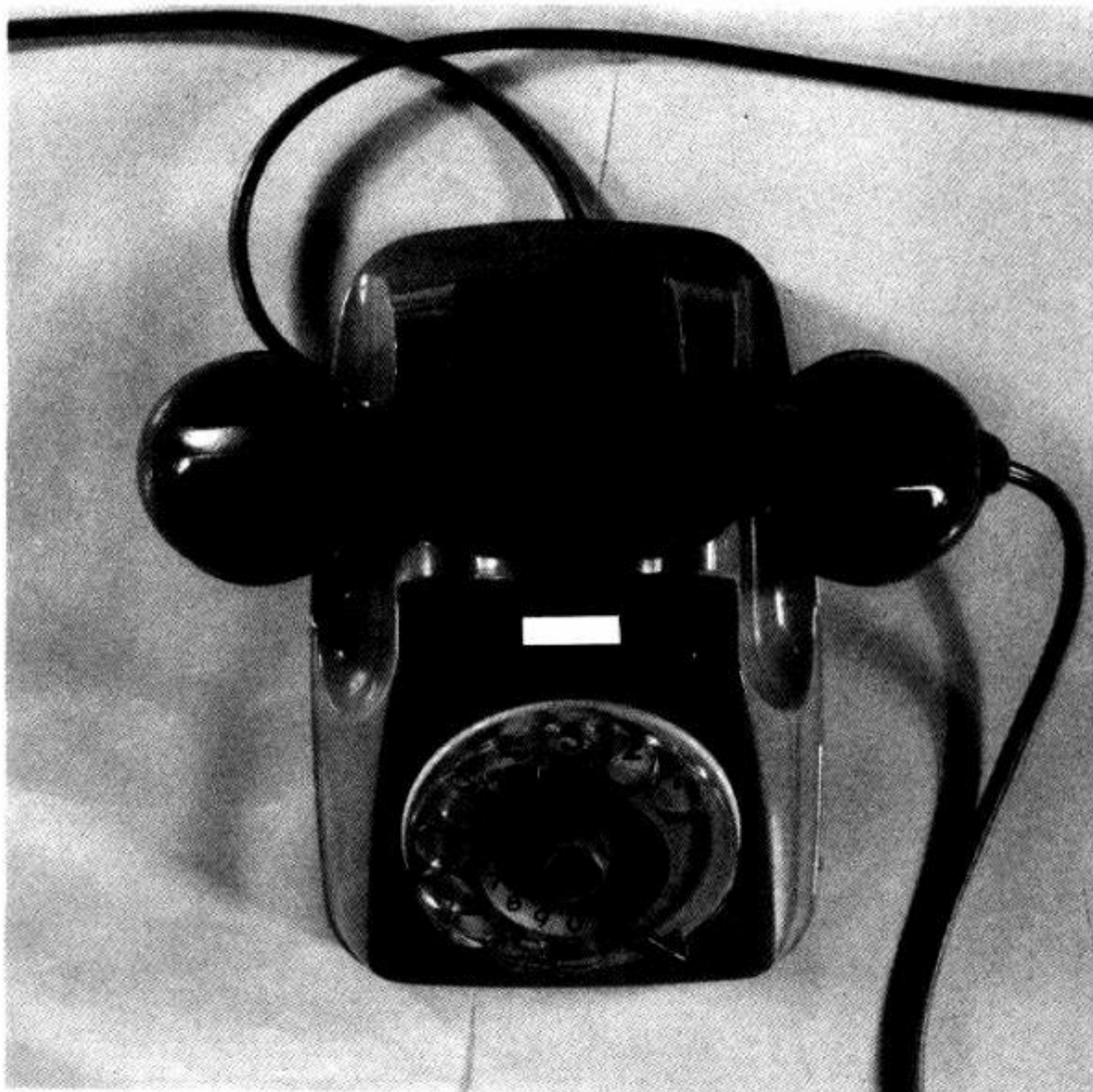
di Antonio Pastorelli

A Positano, il 30 maggio scorso, si è tenuto un convegno centrato sull'accordo recentemente siglato dalle tre aziende nel favorire lo sviluppo della telematica di massa.

La Commodore, forte della cifra di un milione e seicentomila pezzi venduti in Italia (dei quali un milione sono C-64), rappresenta, tra le industrie del settore, l'azienda ideale per raggiungere l'obiettivo prefissato.

La Sip e la Seat, leaders nella fornitura di strumenti per la diffusione dell'informazione, hanno creato, circa un anno fa, il primo servizio telematico pubblico in Italia: il Videotel.

Oggi tale servizio offre agli utenti informazioni attinte da più di 250 banche-dati che soddisfano le più disparate esigenze di potenziali utenti: dai notiziari ai servizi di home-banking ed al telesoftware, dalla prenotazione di alberghi alla consultazione di biblioteche.

Con l'accordo Sip, Seat e Commodore, l'utente privato potrà, con la modica spesa relativa all'acquisto del C-64, disporre di un servizio finora accessibile solo alle grosse imprese.

Fino ad oggi, infatti, l'utenza telematica era costituita quasi esclusivamente da aziende di una certa mole, a causa dell'elevato costo dei servizi e dei terminali idonei a stabilire il "contatto".

Il C-64, strumento informatico personale a basso costo, si trasforma facilmente in un versatile strumento telematico, aperto verso il mondo intero.

A questo proposito la Commodore ha messo a punto un "adattatore telematico", siglato 6499, comprensivo di software di comunicazione su E-prom, ed omologato dal Ministero delle Poste e Telecomunicazioni.

L'apparecchio si inserisce nella porta di espansione del calcolatore (quella normalmente usata per le cartucce) e si collega, mediante un cavo, alla presa del telefono.
Il software di comunicazione, che si attiva automaticamente all'accensione del computer, è già predisposto per il Videotel, ma permette anche collegamenti diretti tra utenti.
Il piano aziendale si concretizza, almeno per ora, con la commercializzazione immediata di due "confezioni telematiche":

Telematica 1: C-64
Registratore 1530
Geos V 1.2
Adattatore 6499

Telematica 2: Drive 1541
Mouse
Adattatore 6499

Entrambe le confezioni sono vendute a lire 499.000+iva; ciò significa, fatti i debiti confronti con i più recenti prezzi di listino Commodore, che l'adattatore 6499 è ceduto quasi gratuitamente.
La Commodore sta immettendo sul mercato, proprio in queste settimane, la bellezza di 40000 confezioni, che conta di vendere entro brevissimo tempo.
Chi possiede già un sistema completo potrà acquistare il solo Modem (ma a partire dal prossimo mese di settembre), ad un prezzo di circa 150000+iva.

Anche per la famiglia

Le tre aziende intendono attuare la diffusione della telematica rivolgendosi prevalentemente alle due fasce, tuttora "inesplorate", di potenziali utenti: la prima è costituita prevalentemente dai giovani; la seconda dalle rispettive famiglie.
I giovani, utilizzatori "storici" dell'arcinoto C-64, rappresentano il "canale" ideale attraverso il quale portare la telematica nelle famiglie.
Informazioni su spettacoli musicali, scuola, giochi, e la possibilità di procurarsi software direttamente da casa (la stessa Commodore offrirà gratuitamente software tramite Videotel), sono servizi rivolti soprattutto ai giovani.

Non mancano, tuttavia, iniziative per gli adulti: sarà possibile prenotare posti in teatro, come pure dare disposizioni alla propria banca, circa numerose operazioni finanziarie, stando seduti comodamente in poltrona ed evitando, quanto meno, ore di coda agli sportelli.

Attualmente gli istituti di credito che offrono il servizio di "home-banking" sono pochi, ma si prevede che, ben presto, si adegueranno molte banche, sia per l'impegno della Banca d'Italia nel sollecitare una celere attuazione del servizio, sia, soprattutto, sulla possibilità concreta di rivolgersi ad un parco-utenza di consistenti dimensioni.
L'iniziativa, come è facile intuire, riscuoterà un enorme successo; basti pensare che dopo circa un anno di vita, e con un target di utilizzo limitato ai soli operatori commerciali, il servizio Videotel fornisce, oggi, informazioni provenienti da 250 banche-dati, per un totale di circa 163.000 pagine di informazioni.
C'e da aspettarsi che, in seguito al maggior numero di utenti e alla loro eterogeneità di esigenze, i servizi offerti saranno in numero sempre maggiore e tratteranno argomenti sempre più diversificati.
A titolo di esempio indichiamo la lista, parziale, di fornitori di informazioni su Videotel:

Lasernet 800, La stampa, Comune di Milano, Ferrovie dello stato, British airways, Università studi Milano, Toro assicurazioni, Cassa di risparmio di Venezia, Regione Campania, agenzia giornalistica Telex, Federcaccia, Postal market, Valtur vacanze, Telehotel.

I costi

Per favorire la diffusione della telematica di massa, l'impegno da parte di Sip e Seat tende a limitare il più possibile il costo dei servizi.
Quest'ultimo, infatti, per l'utente "privato", risulta estremamente contenuto, come si può notare dal seguente pro-memoria:

- *L. 50.000 annue, quale canone di concessione governativa (per le utenze non private è di L. 200.000)*
- *Prezzo di un solo scatto all'atto del collegamento telefonico con il centro Videotel (Tel. n.165).*
- *L. 150 ogni 3 minuti (dalle ore 8 alle 22)*
- *L. 150 ogni 9 minuti di collegamento (il sabato, i giorni festivi e durante le ore notturne).*

In ogni caso l'iniziativa commerciale delle tre aziende telematiche va osservata nella panoramica più ampia di una politica di restyling dell'immagine della Commodore, operazione tipica di un'impresa ormai matura ed affermata nel proprio settore.
In questo contesto rientra anche l'impegno per una più attenta considerazione nei confronti dell'utente finale, con un miglioramento della rete distributiva e di assistenza tecnica.
L'utente acquisterà con fiducia, sapendo di poter contare su una assistenza efficiente e qualificata; in definitiva la Commodore intende stabilire un rapporto più diretto con i propri clienti.
E per decidere rapidamente le misure organizzative per la realizzazione di tale proposito, la Commodore ha tenuto una riunione a Bologna il 2 giugno, nel corso della quale sono state presentate le nuove strategie di mercato che si intendono perseguire a breve termine.

Super routine per C/64

Continua la proposta di routine innovative per l'intramontabile computer della Commodore

a cura di Alessandro de Simone

Magic Sound (24573/24734)

Nei videogame sono contenuti spesso effetti sonori fantastici riproducibili facilmente con la routine di queste pagine: a voi sfogarvi con la fantasia per creare suoni mai uditi da orecchio umano!

Un suono è una vibrazione meccanica caratterizzata da forma d'onda, ampiezza e frequenza. La forma d'onda è ciò che definisce il timbro del suono; per esempio un pianoforte e un violino hanno forme d'onda diverse e proprio per questo motivo riusciamo a distinguerli tra loro.

La frequenza indica invece il numero delle volte in cui una singola forma d'onda si ripete in un secondo: definisce l'altezza del suono, creando distinzione tra un suono grave (minore frequenza) e uno acuto (maggiore frequenza); si misura in hertz (Hz). Se, per esempio, udiamo un suono di 500 Hz, vuol dire che la forma d'onda si ripete 500 volte in un secondo; grazie alla variazione di frequenza è possibile comporre musica, suonando un certo numero di note successive a frequenza diversa. Infine l'ampiezza indica il volume del suono.

La routine proposta da Fabio Sorgato fa variare la frequenza di una forma d'onda scelta dall'utente tra le quattro disponibili (dente di sega, triangolare, quadra, rumore bianco) ad una certa velocità.

Gli effetti ottenibili sono quattro: discesa dalla massima frequenza alla minima, salita dalla minima alla massima, discesa e successiva risalita di una frequenza. Il modo d'utilizzo è il seguente:

SYS XXXX, EF, WF, TM

in cui XXXX, come al solito, rappresenta la locazione d'inizio della routine; "ef" è l'effetto desiderato (numerato tra 0 e 3), "wf" la forma d'onda (compresa tra 0 e 3), "tm" l'intervallo di tempo impostato (compreso tra 1 e 255).

```
1000 PRINTCHR$(147)"MAGIC SOUND"
1010 PRINT:PRINT"SERVE A CREARE
     DEGLI EFFETTI SONORI"
1020 PRINT:PRINT"SYS XXXX,EFF,WF
     ,TEMPO"
1030 PRINT"XXXX=LOCAZIONE INIZIO
     "
1040 PRINT"EFF=EFFETTO (0-3)"
1050 PRINT"WF=FORMA D'ONDA (0-3)
     "
1060 PRINT"TEMPO=VELOCITA' (1-25
     5)"
1070 RETURN
1080 DATA  032,253,174,032,158,1
     83,138,041,252,240,005,162,
     014
1090 DATA  108,000,003,134,251,0
     32,253,174,032,158,183,138,
     041
1100 DATA  252,208,238,232,169,0
     08,010,202,208,252,133,252,
     230
1110 DATA  252,032,253,174,032,1
     58,183,138,240,218,134,253,
     169
1120 DATA  015,141,024,212,169,0
     64,141,005,212,169,128,141,
     006
1130 DATA  212,165,252,141,004,2
     12,165,251,041,002,208,037,
     169
1140 DATA  255,133,254,133,255,1
     65,255,141,001,212,165,254,
     141
```

```
1150 DATA  000,212,056,165,254,2
     29,253,133,254,165,255,233,
     000
1160 DATA  133,255,208,231,165,2
     51,201,001,208,037,169,001,
     133
1170 DATA  254,133,255,165,255,1
     41,001,212,165,254,141,000,
     212
1180 DATA  024,165,254,101,253,1
     33,254,165,255,105,000,133,
     255
1190 DATA  208,231,165,251,201,0
     03,240,182,162,024,169,000,
     157
1200 DATA  000,212,202,208,250,0
     96,-1,24919
```

```
 1 REM DIMOSTRATIVO DI MAGIC S
   OUND
 2 :
 5 PRINTCHR$(147)
10 FOR  K=0  TO  3
20 FOR  X=0  TO  3
25 POKE 54274,15:POKE 54275,8:
   REM DISPONE IL DUTY CYCLE P
   ER L'EFFETTO 2
30 PRINTCHR$(19);"SYS 24573,";
   K;X;5
35 SYS24573,K,X,5:NEXTX,K
40 REM 24573=INDIRIZZO SUGGERI
   TO SU C.C.C.
```

```
          Routine:MAGIC SOUND
     ------------------------------

Origin:$C000

       JSR $AEFD   controlla la virgola
       JSR $B79E   e prende un parametro
       TXA         tra 0 e 255 nel
       AND #$FC    registro X, se e'
       BEQ *C010   maggiore di 3,
*C008  LDX #$0E     ?ILLEGAL QUANTITY
       JMP ($0300)  ERROR,altrimenti
*C010  STX $FB     lo deposita in memoria
       JSR $AEFD   controlla la virgola
       JSR $B79E   e prende un altro
       TXA         parametro che deve
       AND #$FC    essere compreso tra 0
       BNE *C008   e 3,altrimenti errore
       INX         crea un byte con
       LDA #$08    acceso il bit indicato
*C020  ASL A       dall'ultimo parametro,
       DEX
       BNE *C020
       STA $FC     lo deposita in memoria
       INC $FC     e setta il bit 0
       JSR $AEFD   prende la virgola
       JSR $B79E   e il parametro della
       TXA         velocita' che deve
       BEQ *C008   essere diverso da 0,
       STX $FD     lo deposita in $FD.
       LDA #$0F    alza il volume
       STA $D418
       LDA #$40    setta A/D
       STA $D405
       LDA #$80    e S/R
       STA $D406
       LDA $FC     ed infine la forma
       STA $D404   d'onda calcolata prima
       LDA $FB     se e' non e' acceso il
       AND #$02    bit 1
       BNE *C072
*C04D  LDA #$FF    il primo segmento
       STA $FE     sara' in discesa
       STA $FF     dispone la nota piu'
*C053  LDA $FF     alta
       STA $D401   e la suona.

       LDA $FE
       STA $D400
       SEC
       LDA $FE     calcola la nota
       SBC $FD     successiva
       STA $FE
       LDA $FF
       SBC #$00
       STA $FF
       BNE *C053   se non e' la nota piu'
       LDA $FB     bassa, loop.
       CMP #$01    se e' l'effetto 0
       BNE *C097   esce, altrimenti
*C072  LDA #$01    suona un segmento in
       STA $FE     salita, dispone il
       STA $FF     valore minimo
*C078  LDA $FF     suona la nota
       STA $D401
       LDA $FE
       STA $D400
       CLC
       LDA $FE   calcola la successiva
       ADC $FD
       STA $FE
       LDA $FF
       ADC #$00
       STA $FF
       BNE *C078 se non e' l'ultima,
       LDA $FB   ripete, se e' l'effetto
       CMP #$03  tre suona un segmento
       BEQ *C04D in salita,
*C097  LDX #$18    altrimenti azzera
       LDA #$00    i registri da $D400
*C098  STA $9D00,X a $D418
       DEX
       BNE *C098
       RTS         ed esce
```

Sprite mover (24735/24965)

Questa routine, anch'essa di Fabio Sorgato, serve per muovere sullo schermo uno degli otto sprite, nella direzione e con la velocità volute, fermandolo, a scelta, quando si verifica una collisione con un altro sprite o con un carattere e, comunque, quando esce dallo schermo.

Per muovere un proiettile-sprite per mezzo del Basic, come è noto, occorre molto tempo, che si incrementa ulteriormente se, dopo ogni spostamento, desideriamo verificare eventuali collisioni.

Il linguaggio macchina viene ancora una volta in aiuto consentendoci di migliorare notevolmente la qualità del videogame.

La routine per il funzionamento richiede numerosi parametri:

SYS XXXX, numspr, stx, sty, incx, incy, vel, coll

XXXX definisce la locazione d'inizio della nostra routine, rilocabile come qualsiasi altra routine dell'enciclopedia in linguaggio macchina.
NUMSPR è il numero delio sprite (deve essere compreso tra 0 e 7)
STX è la coordinata X di partenza dello sprite (compresa tra 0 e 512).
STY è la coordinata Y (tra 0 e 200).
INCX è l'incremento voluto sull'asse delle X, deve essere compreso tra 0 e 127 e può essere positivo o negativo: se negativo è il numero complementato a 2 (vedi più avanti per maggiori dettagli).
INCY come INCX, solo che vale per l'asse Y.
VEL, indica la velocità di spostamento dello sprite (tra 0 e 255)
COLL, indica il tipo di collisione che determina l'uscita dalla routine; può essere:
0=collisione sprite-sprite.
1=collisione sprite-carattere.
Se si verifica una collisione, nella locazione 2 verrà messo il valore 1, altrimenti sarà sempre presente il valore 0.

Complementare a 2 un numero, per effettuare una somma algebrica binaria, vuol dire cambiare di stato tutti i bit (gli 0 diventano 1 e gli 1 diventano 0) e sommare, al byte così ottenuto, il valore 1 (bit 0 a 1, tutti gli altri a 0).

Se, per esempio, vogliamo che l'incremento X sia -1 dovremo mettere al parametro INCX 256-1 (=255), se -10 sarà 256-10 e così via.

Per la velocità si consiglia di usare parametri superiori a 5, perchè se la velocità di movimento risulta più elevata di quella di scansione video può capitare che la collisione non venga rilevata.

```
1000 PRINTCHR$(147)"SPRITE MOVER
     "
1010 PRINT:PRINT"MUOVE UNO SPRIT
     E SU SCHERMO"
1020 PRINT"RILEVANDONE UN'EVENTU
     ALE COLLISIONE"
1030 PRINT"CON UN CARATTERE O UN
     'ALTRO SPRITE"
1040 PRINT:PRINT"SYS XXXX,NUMSP,
     X,Y,INCX,INCY,VEL,COLL"
1050 PRINT:PRINT"XXXX=LOCAZIONE
     INIZIO"
1060 PRINT"NUMSP=NUMERO SPRITE (
     0-7)"
1070 PRINT"X,Y=PUNTO DI PARTENZA
      SPRITE"
1080 PRINT"INCX=INCREMENTO ASSE
     X"
1090 PRINT"INCY=INCREMENTO ASSE
     Y"
1100 PRINT TAB(5)"I VALORI NEGAT
     IVI DI INCX E INCY"
1110 PRINT TAB(5)"DEVONO ESSERE
     COMPLEMENTATI A 2"
1120 PRINT TAB(5)"PER ESEMPIO IN
     CX=-5"
1130 PRINT TAB(5)"DIVENTA INCX=2
     56-5"
1140 PRINT TAB(5)"INCY=-20 DIVEN
     TA INCY=256-20"
1150 PRINT"VEL=VELOCITA' (0-255)
1160 PRINT"COLL=COLLISIONE 0=CON
      SPRITE 1=CON CAR.
1170 PRINT"SE C'E' COLLISIONE LA
      LOCAZIONE 2"
1180 PRINT"VERRA' MESSA A 1,ALTR
     IMENTI A 0"
1190 RETURN
1200 DATA 032,241,183,138,041,2
     48,240,005,162,014,108,000,
     003
1210 DATA 134,251,032,253,174,0
     32,235,183,165,020,141,160,
     002
1220 DATA 165,021,141,161,002,1
     42,162,002,032,241,183,134,
     252
1230 DATA 032,241,183,134,253,0
     32,241,183,134,254,032,241,
     183
1240 DATA 138,041,254,208,207,1
     34,255,169,001,166,251,240,
     004
1250 DATA 010,202,208,252,141,1
     63,002,173,030,208,173,031,
     208
```

```
1260 DATA  165,251,010,168,173,1
     60,002,153,000,208,173,162,
     002
1270 DATA  153,001,208,173,161,0
     02,240,008,173,163,002,141,
     016
1280 DATA  208,208,011,173,163,0
     02,073,255,045,016,208,141,
     016
1290 DATA  208,165,255,208,010,1
     73,030,208,045,163,002,240,
     018
1300 DATA  208,008,173,031,208,0
     45,163,002,240,008,169,001,
     133
1310 DATA  002,096,024,144,186,1
     64,254,162,255,202,208,253,
     136
1320 DATA  208,250,173,162,002,0
     24,101,253,141,162,002,165,
     253
1330 DATA  041,128,240,004,144,0
     51,176,002,176,047,173,160,
     002
1340 DATA  024,101,252,141,160,0
     02,165,252,041,128,208,019,
     144
1350 DATA  003,238,161,002,173,1
     61,002,240,197,173,160,002,
     201
1360 DATA  114,144,190,176,013,1
     76,186,056,173,161,002,233,
     001
1370 DATA  141,161,002,176,175,1
     69,000,133,002,096,-1,29904
```

```
  1 REM DIMOSTRATIVO DI SPRITE
    MOVER
  2 :
  5 INPUT "VELOCITA' (1=MAX 255
    =MIN)";V
 10 FOR K=832 TO 895:POKE K,255
    :NEXT:REM  RIEMIE SPRITE
 20 POKE 2040,13:POKE 53269,1:R
    EM  SCEGLIE BANCO/ACCENDE S
    PRITE
 21 POKE 53287,1:POKE 53275,1:R
    EM  COLORE E PRIORITA SPRIT
    E/FONDO
 24 PRINT CHR$(147) TAB(97)"*"
 25 PRINT  TAB(255)"PIPPO"
 30 SYS 24735,0,350,0,255,1,V,1
    :GOSUB  100
 40 SYS 24735,0,0,200,1,255,V,1
    :GOSUB  100
 50 FOR K=0 TO 16:POKE 53287,K:
    FOR X=1 TO 50:NEXT:NEXT
 60 PRINTCHR$(19)"              "C
    HR$(19) TAB(97)" "
 70 SYS24735,0,138,62,0,1,V*2,1
    :GOSUB 100
 90 END
100 IF  PEEK(2)=1  THEN PRINTCH
    R$(19)"COLPITO":RETURN
110 PRINTCHR$(19)"MANCATO":RETU
    RN
```

```
      JSR $B7F1    controlla la virgola
      TXA          e prende un parametro
      AND #$F8     in X, che deve essere
      BEQ *C00D    compreso tra 0 e 7
*C008 LDX #$0E     altrimenti ILLEGAL
      JMP ($0300)    QUANTITY ERROR
*C00D STX $FB      lo deposita
      JSR $AEFD    controlla una virgola
      JSR $B7EB    prende due parametri,
      LDA $14      uno a 16 bit in
      STA $02A0    $14-$15 e uno a 8 bit
      LDA $15      in X, separati da
      STA $02A1    una virgola e li
      STX $02A2    deposita
      JSR $B7F1    prende un parametro
      STX $FC      separato da virgola
      JSR $B7F1    idem
      STX $FD
      JSR $B7F1    idem
      STX $FE
      JSR $B7F1    idem e controlla
      TXA          che sia 0 o 1
      AND #$FE
      BNE *C008    altrimenti errore
      STX $FF
      LDA #$01     accende il bit
      LDX $FB      indicato dal byte
      BEQ *C045    $FB
*C041 ASL A
      DEX
      BNE *C041    e deposita il
*C045 STA $02A3    risultato
      LDA $D01E    azzera i registri di
      LDA $D01F    controllo collisione.
*C04E LDA $FB      raddoppia il numero
      ASL A        dello sprite e lo
      TAY          usa per sistemare
      LDA $02A0    le coordinate di
      STA $D000,Y  partenza.
      LDA $02A2
      STA $D001,Y
      LDA $02A1    se le X sono superi-
      BEQ *C06B    ori a 255
      LDA $02A3
      STA $D010    accende il MSB
      BNE *C076
*C06B LDA $02A3    altrimenti lo spegne
      EOR #$FF
      AND $D010
      STA $D010
*C076 LDA $FF      se il parametro
      BNE *C084    indica 1, controlla
      LDA $D01E    la collisione
      AND $02A3    sprite-sprite
      BEQ *C094    e se riscontrata esco
      BNE *C08C
*C084 LDA $D01F    altrimenti controlla
      AND $02A3    la collisione
      BEQ *C094    sprite-carattere,
*C08C LDA #$01     se la riscontra esce
      STA $02
      RTS
*C091 CLC          (ritorna daccapo)
      BCC *C04E
*C094 LDY $FE      altrimenti attende in
      LDX #$FF     relazione alla
*C098 DEX          velocita' impostata
      BNE *C098
      DEY
      BNE *C098
      LDA $02A2    somma alle Y il para-
      CLC          metro,
      ADC $FD
      STA $02A2
      LDA $FD      se era negativo e
      AND #$80
      BEQ *C0B1
      BCC *C0E2    il risultato <0 esce,
      BCS *C0B3    altrimenti continua
*C0B1 BCS *C0E2    se positivo e il ri-
                   sultato >255 esce
*C0B3 LDA $02A0    altrimenti somma
      CLC          il parametro alle X,
      ADC $FC
      STA $02A0    se il parametro era
      LDA $FC      positivo
      AND #$80
      BNE *C0D5
      BCC *C0C7    e la somma >255 mette
      INC $02A1    a 1 il byte alto;
*C0C7 LDA $02A1    se le coordinate
      BEQ *C091    sono inferiori a
      LDA $02A0    $172 (fuori schermo)
      CMP #$72
      BCC *C091    riprende daccapo,
      BCS *C0E2    altrimenti esce.
                   se il parametro era
*C0D5 BCS *C091    negativo e il risul-
      SEC          tato <0,
      LDA $02A1    decrementa il byte
      SBC #$01     alto,
      STA $02A1
      BCS *C091    se >0 loop,altrimenti
*C0E2 LDA #$00     azzera il byte $02
      STA $02
      RTS          ed esce
```

Hard/Copy Hi-Res (24966/25374) (Richiede MPS/803 o compatibili)

Questa routine di Maurizio Dell'Abate serve per stampare su carta una pagina grafica in alta risoluzione (320*200 dots). Una routine di questo tipo non è certo cosa inedita, tuttavia quella che vi proponiamo si differenzia dalle altre perchè è *completamente rilocabile*: ciò significa che funziona correttamente qualunque sia l'indirizzo di partenza (del resto, la rilocabilità caratterizza tutte le routine di questa rubrica).

L'uso è semplicissimo: prima della chiamata, che può avvenire mediante SYS (dal Basic) o JSR (da L.M.), si deve comunicare al programma la posizione in memoria degli 8000 byte consecutivi che rappresentano la pagina in alta risoluzione. Basterà immettere poi, nella locazione 170 ($AA), l'indirizzo del primo byte della pagina grafica, diviso per 256.

Se, per esempio, la pagina inizia in 8192 (come nella maggior parte dei casi), digiteremo:

POKE 170,32: SYS (indirizzo di partenza)

Il valore 32, come i più acuti(!) avranno notato, non è altro che la 256esima parte di 8192.

E' ovvio che al momento della chiamata la stampante deve essere accesa, correttamente collegata e munita di carta, pena disperazione dell'utente distratto. La routine, nonostante gli artifici per non compromettere la rilocabilità, risulta piacevolente veloce. Va precisato che, durante la stampa, il computer può trovarsi anche in modo testo.

Esaminare il disassemblato non è sicuramente cosa semplice: le subroutine e i salti incondizionati vengono simulati da istruzioni a salto relativo e, dato che queste hanno una "falcata" limitata, vengono concatenate paurosamente tra loro (leggi: ci stiamo attrezzando per scrivere programmi da 64K rilocabili).

Il Demo di queste pagine traccia sullo schermo una circonferenza e la stampa. Dal momento che la stampante "schiaccia" leggermente le immagini, la circonferenza disegnata sul video è più alta che larga. Dal Demo potete "estrarre", se vi è utile, la brevissima routine per cancellare la pagina grafica.

```
1000 PRINTCHR$(147);"HARD COPY"
1010 PRINT"QUESTA ROUTINE SERVE
     PER STAMPARE SU"
1020 PRINT"CARTA UNA SCHERMATA I
     N ALTA RISOLUZIO-"
1030 PRINT"NE (320*200). USO:
1040 PRINT"POKE170,X:SYSY"
1050 PRINT"DOVE X E' L'INIDIRIZZ
     O DELLA PRIMA LO-"
1060 PRINT"CAZIONE DELLA PAGINA
     HIRES DIVISO"
1070 PRINT"PER 256 E Y E' L'INDI
     RIZZO DI PARTENZA"
1080 PRINT"DI QUESTA ROUTINE (MP
     S/803 E COMPAT.)."
1090 RETURN
1100 DATA  032,231,255,169,000
1110 DATA  032,189,255,169,001
1120 DATA  162,004,160,000,032
1130 DATA  186,255,032,192,255
1140 DATA  162,001,032,201,255
1150 DATA  169,013,032,210,255
1160 DATA  169,008,032,210,255
1170 DATA  169,000,141,172,002
1180 DATA  169,000,133,251,133
1190 DATA  252,024,144,004,234
1200 DATA  234,208,243,173,172
1210 DATA  002,010,141,169,002
1220 DATA  010,024,109,172,002
1230 DATA  109,169,002,133,253
1240 DATA  162,000,142,170,002
1250 DATA  142,171,002,134,254
1260 DATA  162,003,181,251,072
1270 DATA  202,016,250,024,144
1280 DATA  004,234,234,208,214
1290 DATA  165,251,041,007,141
1300 DATA  167,002,169,007,056
1310 DATA  237,167,002,141,167
1320 DATA  002,165,253,041,007
1330 DATA  141,168,002,162,003
1340 DATA  070,252,102,251,070
1350 DATA  254,102,253,202,208
1360 DATA  245,165,253,133,168
1370 DATA  165,254,133,169,162
1380 DATA  008,006,253,038,254
1390 DATA  202,208,249,024,144
1400 DATA  004,208,154,208,181
1410 DATA  162,006,006,168,038
1420 DATA  169,202,208,249,024
1430 DATA  165,253,101,168,133
1440 DATA  253,165,254,101,169
1450 DATA  133,254,162,003,006
1460 DATA  251,038,252,202,208
1470 DATA  249,024,165,251,101
1480 DATA  253,133,251,165,252
1490 DATA  101,254,133,252,024
1500 DATA  144,004,208,205,208
1510 DATA  143,165,252,101,170
1520 DATA  133,252,024,165,251
1530 DATA  109,168,002,133,251
1540 DATA  165,252,105,000,133
1550 DATA  252,174,167,002,208
1560 DATA  008,169,001,141,167
```

```
1570 DATA  002,024,144,020,169
1580 DATA  002,141,167,002,224
1590 DATA  001,240,011,014,167
1600 DATA  002,202,024,144,245
1610 DATA  208,150,208,201,160
1620 DATA  000,177,251,045,167
1630 DATA  002,133,002,162,000
1640 DATA  104,149,251,232,224
1650 DATA  004,208,248,165,002
1660 DATA  240,004,169,001,133
1670 DATA  002,174,170,002,240
1680 DATA  005,006,002,202,208
1690 DATA  251,165,002,013,171
1700 DATA  002,141,171,002,230
1710 DATA  253,238,170,002,173
1720 DATA  170,002,201,007,208
1730 DATA  142,173,171,002,009
1740 DATA  128,141,171,002,024
1750 DATA  144,004,234,234,208
1760 DATA  180,174,172,002,224
1770 DATA  028,240,007,162,255
1780 DATA  134,002,024,144,004
1790 DATA  162,143,134,002,173
1800 DATA  171,002,037,002,032
1810 DATA  210,255,024,165,251
1820 DATA  105,001,133,251,165
1830 DATA  252,105,000,133,252
1840 DATA  201,001,208,139,165
1850 DATA  251,201,064,208,133
1860 DATA  238,172,002,169,013
1870 DATA  032,210,255,173,172
1880 DATA  002,201,029,208,190
1890 DATA  169,015,032,210,255
1900 DATA  169,013,032,210,255
1910 DATA  032,231,255,096
1920 DATA  -1,54555
```

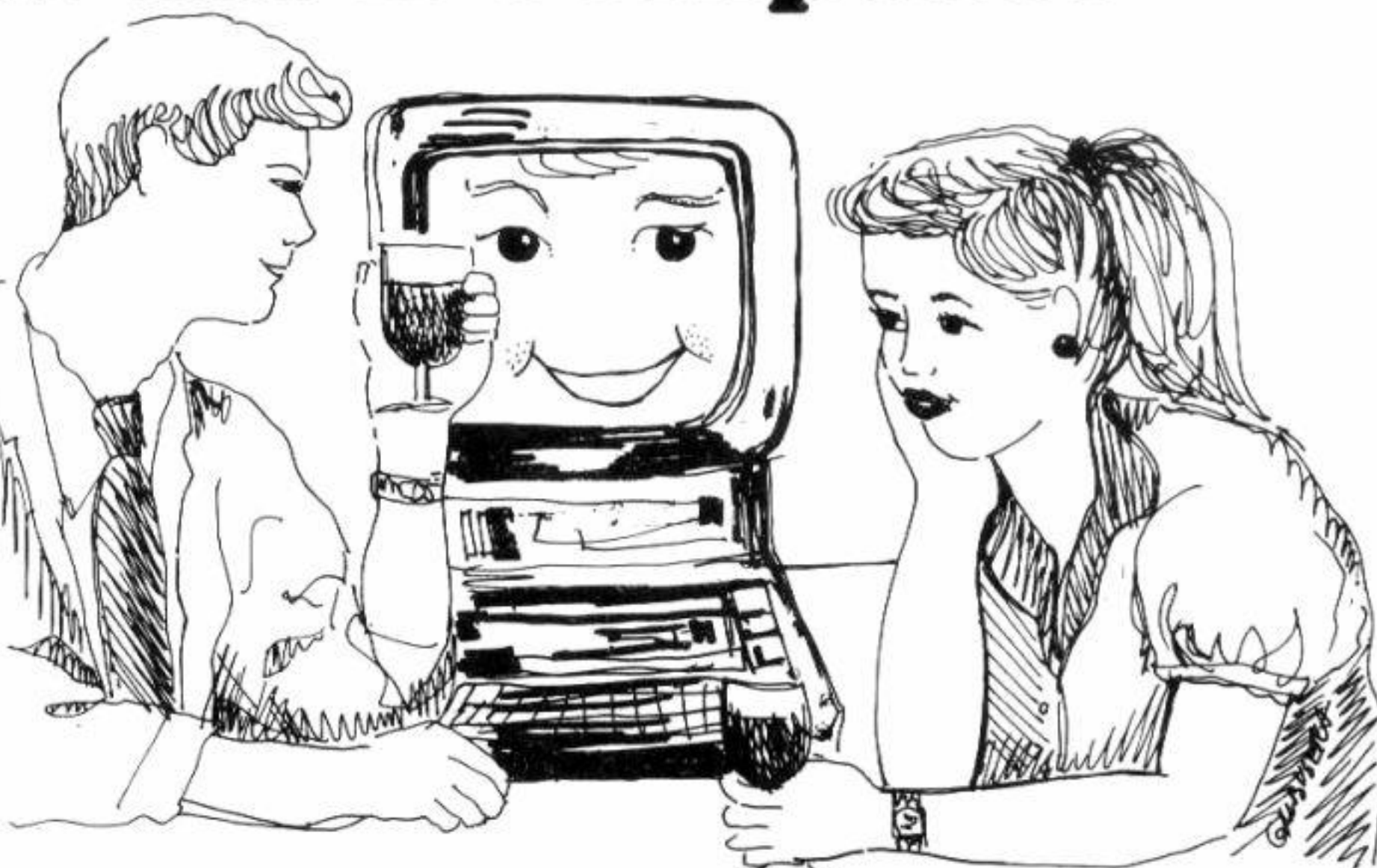
```
100 REM  DEMO HCOPY PER C/64
110 :
120 X=24966:REM  INDIRIZZO SUGG
    ERITO DA C.C.C.
130 Z=53272:C=53265
140 POKE 53265,PEEK(53265) OR 3
    2:POKE 53272,PEEK(53272) OR
     8:PRINTCHR$(147)
150 FOR I=0 TO 30:READ A:POKE 6
    79+I,A:NEXT:SYS679:REM  PUL
    ISCE LA PAGINA GRAFICA
160 FOR CV=.01 TO 2*π STEP .04
170 O=SIN(CV)*80+160:Y=COS(CV)*
    95+100
180 BY=8192+INT(Y/8)*320+INT(O/
    8)*8+(Y AND 7):POKE BY,PEEK
    (BY) OR 2↑(7-(O AND 7))
190 NEXT:POKE 53265,PEEK(53265)
     AND 223:POKE 53272,21:PRIN
    TCHR$(147):POKE 170,32
200 PRINT"PREPARA LA STAMPANTE
    E PREMI UN TASTO!":POKE 198
    ,0:WAIT 198,1:POKE 198,0
210 SYSX:REM  HARD COPY
220 DATA 169,0,133,251,169,32,1
    33,252,160,0,152,145,251,24
    ,165,251,105,1
230 DATA 133,251,165,252,105,0,
    133,252,201,64,208,234,96:R
    EM  DATA PER CLEAR
```

```
2000 JSR $ffe7 chiude tutti i files.
2003 LDA #$00  nome del file in
2005 JSR $ffbd apertura nullo.
2008 LDA #$01  numero del file.
200a LDX #$04  numero di device.
200c LDY #$00  indirizzo secondario.
200e JSR $ffba imposta i tre valori.
2011 JSR $ffc0 apre il file.
2014 LDX #$01  considera il file 1
2016 JSR $ffc9 come output.
2019 LDA #$0d  effettua un
201b JSR $ffd2 carriage return.
201e LDA #$08  manda la stampante
2020 JSR $ffd2 in alta risoluzione.
2023 LDA #$00  azzera il
2025 STA $02ac contatore.
2028 LDA #$00  azzera alcuni
202a STA $fb   puntatori usati
202c STA $fc   dalla routine.
202e CLC       salto incondizionato
202f BCC $2035 a $2035.
2031 NOP       nessuna operazione.
2032 NOP       nessuna operazione.
2033 BNE $2028 salta a $2028 se z=0.
2035 LDA $02ac queste istruzioni
2038 ASL       servono per moltiplicare
2039 STA $02a9 per 7 il valore
203c ASL       contenuto nella
203d CLC       locazione $02ac, il
203e ADC $02ac risultato viene messo
2041 ADC $02a9 nella locazione di
2044 STA $fd   memoria $fd.
2046 LDX #$00  azzera
2048 STX $02aa le locazioni
204b STX $02ab di memoria
204e STX $fe   $02aa, $02ab, $fe.
2050 LDX #$03  questa parte della
2052 LDA $fb,x routine salva nello
2054 PHA       stack il contenuto
2055 DEX       delle locazioni
2056 BPL $2052 $fb, $fc, $fd, $fe.
2058 CLC       esegue un salto in-
2059 BCC $205f condizionato a $205f.
205b NOP       nessuna operazione.
205c NOP       nessuna operazione.
205d BNE $2035 salta a $2035 se z=0.
205f LDA $fb   effettua un AND logico
2061 AND #$07  tra $fb e #$07 e pone
2063 STA $02a7 il risultato in $02a7.
2066 LDA #$07  sottrae il contenuto di
2068 SEC       $02a7 dal valore #$07,
2069 SBC $02a7 pone il risultato in
```

```
206c STA $02a7 $02a7 stessa.
206f LDA $fd   effettua un AND logico
2071 AND #$07  tra $fd e #$07, il ri-
2073 STA $02a8 sultato va in $02a8.
2076 LDX #$03  queste istruzioni
2078 LSR $fc   dividono per 8 i due
207a ROR $fb   valori posti nelle loca-
207c LSR $fe   zioni $fb-$fc e $fd-$fe,
207e ROR $fd   secondo il tradizionale
2080 DEX       formato byte basso -
2081 BNE $2078 byte alto.
2083 LDA $fd   pone il contenuto
2085 STA $a8   di $fd ed $fe rispetti-
2087 LDA $fe   vamente nelle locazioni
2089 STA $a9   $a8 e $a9.
208b LDX #$08  moltiplica il valore
208d ASL $fd   byte basso - byte alto
208f ROL $fe   contenuto in $fd-$fe
2091 DEX       per 256. il risultato
2092 BNE $208d resta in tali locazioni.
2094 CLC       esegue un salto incon-
2095 BCC $209b dizionato a $209b.
2097 BNE $2033 salta a $2033 se z=0.
2099 BNE $2050 salta a $2050 se z=0.
209b LDX #$06  moltiplica il valore
209d ASL $a8   byte basso - byte alto
209f ROL $a9   contenuto in $a8-$a9
20a1 DEX       per 64. il risultato
20a2 BNE $209d resta in tali locazioni.
20a4 CLC       somma tra loro i valori
20a5 LDA $fd   byte basso - byte alto
20a7 ADC $a8   contenuti nelle due cop-
20a9 STA $fd   pie di locazioni
20ab LDA $fe   $a8-$a9, $fd-$fe. il ri-
20ad ADC $a9   sultato va nella coppia
20af STA $fe   $fd-$fe.
20b1 LDX #$03  moltiplica per
20b3 ASL $fb   8 il valore contenuto
20b5 ROL $fc   nella coppia
20b7 DEX       di bytes $fb-$fc, il ri-
20b8 BNE $20b3 sultato in tali bytes.
20ba CLC       somma tra di loro i
20bb LDA $fb   valori a due bytes
20bd ADC $fd   (basso-alto) contenuti
20bf STA $fb   nelle coppie $fb-$fc e
20c1 LDA $fc   $fd-$fe e pone il risul-
20c3 ADC $fe   tato nella coppia
20c5 STA $fc   $fb-$fc.
20c7 CLC       esegue un salto incondi-
20c8 BCC $20ce zionato a $20ce.
20ca BNE $2099 salta a $2099 se z=0.
20cc BNE $205d salta a $205d se z=0.
20ce LDA $fc   offset per la
20d0 ADC $aa   posizione in memoria
20d2 STA $fc   della pagina grafica.
20d4 CLC       somma il valore a due
20d5 LDA $fb   bytes contenuto nella
20d7 ADC $02a8 coppia $fb-$fc con il
20da STA $fb   valore a un
20dc LDA $fc   solo byte contenuto
20de ADC #$00  in $02a8. il risultato
20e0 STA $fc   viene posto in $fb-$fc.
20e2 LDX $02a7 salta a $20ef solo se
20e5 BNE $20ef $02a7 e' diverso da 0.
20e7 LDA #$01  pone il valore unitario
20e9 STA $02a7 in $02a7 ed esegue
20ec CLC       un salto incondizionato
20ed BCC $2103 a $2103.
20ef LDA #$02  queste istruzioni
20f1 STA $02a7 effettuano un elevamento
20f4 CPX #$01  a potenza. base = #$02;
20f6 BEQ $2103 l'esponente e' contenuto
20f8 ASL $02a7 nel registro x ed il
20fb DEX       risultato viene posto
20fc CLC       in $02a7. salto incondi-
20fd BCC $20f4 zionato a $20f4.
20ff BNE $2097 salta a $2097 se z=0.
2101 BNE $20cc salta a $20cc se z=0.
2103 LDY #$00    legge lo stato di un
2105 LDA ($fb),y dot della pagina gra-
2107 AND $02a7   fica e, se spento, po-
210a STA $02   ne 0 in locazione $02.
210c LDX #$00  recupera i bytes
210e PLA       $fb, $fc, $fd, $fe
210f STA $fb,x precedentemente
2111 INX       salvati
2112 CPX #$04  nello
2114 BNE $210e stack.
2116 LDA $02   salta a $211e solo se
2118 BEQ $211e $02 contiene 0 (zero).
211a LDA #$01  pone il valore unitario
211c STA $02   nella locazione $02.
211e LDX $02aa salta a $2128 solo se
2121 BEQ $2128 $02aa contiene 0 (zero).
2123 ASL $02   shifta a sinistra $02
2125 DEX       per un numero di volte
2126 BNE $2123 pari al reg. x.
2128 LDA $02   effettua un or logico
212a ORA $02ab tra $02ab e $02, pone
212d STA $02ab il risultato in $02ab.
2130 INC $fd   incrementa i contatori
2132 INC $02aa e, se $02aa e' minore
2135 LDA $02aa di 7, salta alla
2138 CMP #$07  locazione $20ca, altri-
213a BNE $20ca menti continua da $213c.
213c LDA $02ab la locazione $02ab con-
213f ORA #$80  tiene il byte "grafico"
2141 STA $02ab da mandare alla printer.
2144 CLC       salto incondizionato
2145 BCC $214b a $214b.
2147 NOP       nessuna operazione.
2148 NOP       nessuna operazione.
2149 BNE $20ff salta a $20ff se z=0.
214b LDX $02ac salta a $2159 se manca
214e CPX #$1c  solo la parte piu' bassa
2150 BEQ $2159 del video da stampare.
2152 LDX #$ff  pone #$ff in $02 ed
2154 STX $02   esegue un salto
2156 CLC       incondizionato a
2157 BCC $215d $215d.
2159 LDX #$8f  pone #$8f nella
215b STX $02   locazione $02.
215d LDA $02ab immette nel buffer
2160 AND $02   della printer il byte
2162 JSR $ffd2 grafico da stampare.
2165 CLC       incrementa le
2166 LDA $fb   coordinate orizzontali
2168 ADC #$01  per la scansione
216a STA $fb   della pagina
216c LDA $fc   grafica, e,
216e ADC #$00  se non si e'
2170 STA $fc   ancORA giunti
2172 CMP #$01  nella parte piu'
2174 BNE $2101 a destra del video,
2176 LDA $fb   il programma
2178 CMP #$40  continua dalla
217a BNE $2101 locazione $2101.
217c INC $02ac incrementa il contatore.
217f LDA #$0d  nuova riga: manda un
2181 JSR $ffd2 return alla stampante.
2184 LDA $02ac se il lavoro non e'
2187 CMP #$1d  terminato, salta
2189 BNE $2149 a $2149.
218b LDA #$0f  fine lavoro:
218d JSR $ffd2 la stampante viene
2190 LDA #$0d  posta in modo testo,
2192 JSR $ffd2 tutti i files vengono
2195 JSR $ffe7 chiusi.
2198 RTS       ritorna.
```

Lui, lei e... il computer

Alcune considerazioni su quello che potrebbe essere il "triangolo perfetto" degli anni '80

di Michele Maggi

Il dramma della gelosia, il terrore del tradimento e dell'infedeltà.

Quante parole spese per commentare, analizzare e cercare di spiegare fenomeni, tipicamente umani, che ci accompagnano fin dalla notte dei tempi.

Nonostante questa introduzione, che potrebbe sembrare fuori luogo sulle pagine di C.C.C, vi assicuro che di questo problema esiste anche l'aspetto informatico: vediamo perchè.

Chiunque abbia il "pallino" del computer e, oltre al suddetto pallino, anche una fidanzata/moglie/amica (o altre creature sentimentalmente vicine, per non parlare delle madri), si sarà trovato più volte coinvolto, suo malgrado, in interminabili discussioni circa l'uso e l'abuso di queste "macchine infernali", discussioni che, col passare del tempo, tendono a degenerare e ad assumere toni sempre più drammatici fino ad arrivare a situazioni di incomprensione decisamente fuori luogo rispetto alla reale dimensione del problema.

In America, Paese da sempre all'avanguardia in stranezze e novità di ogni genere, da un po' di anni a questa parte è iniziata la "moda" delle separazioni a causa del computer, dovute al fatto che i mariti troppo spesso trascurerebbero le mogli preferendo dedicarsi al computer.

Certo, dedicarsi al computer assorbe tempo notevole: è un dato di fatto incontestabile; ma da qui al divorzio...

Considerando che il "terrore" della donna è sempre stato quello "dell'altra", il pensare che il computer possa essere oggetto di gelosie fa senz'altro sorridere.

Tuttavia l'argomento, anche se a prima vista può sembrare ridicolo, assume una certa rilevanza in quanto investe una sfera importante della vita della "coppia informatica".

In primo luogo chi si dedica con passione (per motivi professionali e non) ai computer, raramente si rende conto di quanto il tempo passi in fretta e, in secondo luogo, non sempre è facile "staccarsi" quando si sta svolgendo un lavoro importante che presuppone una certa concentrazione.

Tutto ciò risulta troppo spesso di difficile comprensione per le "divine creature" che, dopo un breve periodo di pseudo-interessamento al lavoro del "fidanzato/marito/amico", tendono a perdere la pazienza, con le conseguenze di cui sopra.

Chi ha ragione? Ma soprattutto, chi ha torto?

Una domanda molto difficile *(che rivolgiamo ai nostri lettori, pregandoli di inviarci il loro parere)* che non sembra avere un'unica risposta; è la tipica situazione in cui entrambe la parti hanno una dose di ragione.

Ai "computermaniaci" forse converrebbe limitare il tempo dedicato al computer cercando di mettersi nei panni delle rispettive "metà" e cercare di immaginare che cosa succederebbe se fossero costretti a dover "subire" passatempi tipicamente femminili come shopping, sfilate di moda e così via (non ce ne vogliano le femministe).

Alle "fidanzate/mogli/amiche" forse converrebbe cercare di capire quali aspetti del computer attirano il loro partner in modo da tentare di avvicinarsi il più possibile alla macchina che, a dispetto di ciò che si sente dire, non è nè arida nè vuota.

La soluzione prospettata certamente non brilla per novità (il compromesso è vecchio come il mondo) ma, se gestita con intelligenza, può portare sicuramente ad una sorta di "triangolo" che fortunatamente(?) ha molto poco a che vedere con i "triangoli" tradizionali.

Amante, computer o... nessuno dei due?

Come "truccare" un joystick

Un chip, quattro pile e un paio di saldature per divertirsi e imparare qualcosa di nuovo

di Marco Ciatti & Lorenzo Barazzetta

In alcuni videogiochi è necessario usare ripetutamente il pulsante di fuoco (più noto con il termine FIRE) del joystick per difendersi dagli attacchi nemici, con probabili conseguenze per il vostro pollice (o indice).

Vi proponiamo, quindi, un semplicissimo progetto hardware che permetterà di aumentare l'efficienza delle vostre incursioni negli scenari dei videogame.

Principio generale di funzionamento

Il computer può individuare lo "stato" del pulsante di fuoco (premuto o meno) tramite la lettura di un bit di una certa locazione di memoria. Il bit di cui stiamo parlando assume il valore unitario, oppure nullo, nel caso in cui il tasto sia rispettivamente premuto o rilasciato.

Il bit corrisponde ad uno dei nove pin della porta giochi (vedi schema nel libretto di istruzioni) ed è in grado di assumere i seguenti valori:

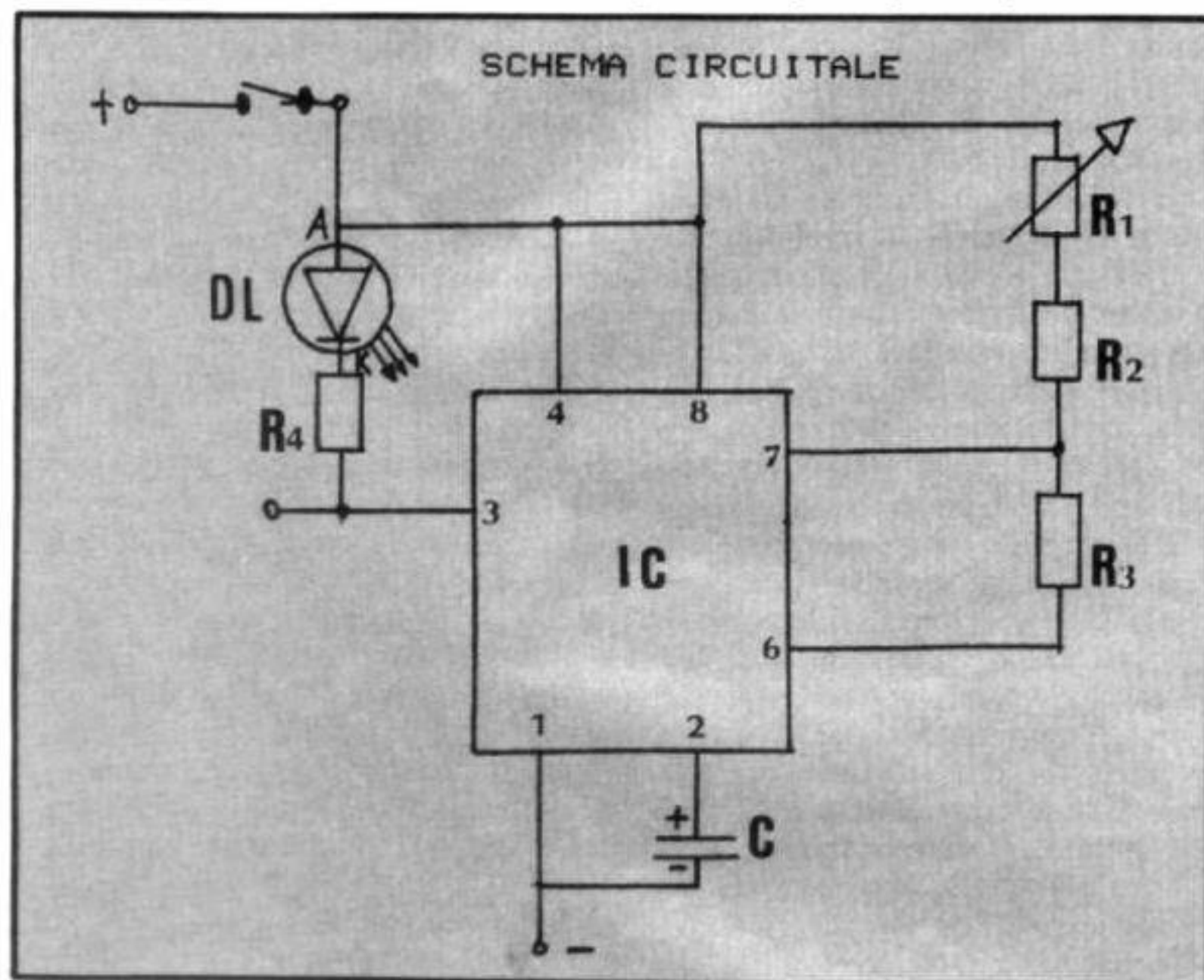
5 Volt (stato logico 1)
0 Volt (stato logico 0)

La pressione del pulsante di fuoco avrà, quindi, la funzione di imporre lo stato logico zero collegando direttamente a massa tale pin.

Lo stesso scopo può essere raggiunto da un circuito elettronico in grado di imporre, alternativamente, i due stati logici (0/1) sostituendo, di fatto, l'azione del malcapitato pollice.

Componenti impiegati nel circuito

Il circuito si serve di un integrato e pochi altri componenti di facile reperibilità.

Elenco componenti:
R1 = potenziometro lineare da 470.000 ohm
R2 = Resistenza da 1.000 ohm
R3 = Resistenza da 1.000 ohm
R4 = Resistenza da 1.800 ohm
C = Condensatore da 10 microfarad 25 volt (elettrolitico radiale)
DL = Diodo led di uso comune
IC = Circuito integrato NE-555
Zoccolo da 8 piedini (per integrati)
Interruttore di uso comune
Contenitore per 4 pile (stilo) da 1.5 Volt

Per chi è digiuno

La resistenza:

Nei circuiti elettronici è necessario dosare assai spesso la corrente elettrica. Si può affermare che, in certi casi, è necessario l'impiego di "cattivi" conduttori di elettricità. Occorrono insomma, lungo i percorsi della corrente elettrica, alcune "porte" di sbarramento parziale (le resistenze) che permettono di dosare a piacimento l'intensità della corrente elettrica.

DIODO LED

A K

Il condensatore:

Nei circuiti elettronici è necessario talvolta immagazzinare cariche elettriche per utilizzarle successivamente. Il condensatore assolve questo scopo.

Il diodo:

E' un particolare componente elettronico in grado di consentire il passaggio della corrente elettrica in un solo senso (come i sensi unici delle strade). Precisamente il diodo impiegato nel circuito è un Led, che si illumina se attraversato dalla corrente nel verso giusto.

L'integrato NE-555:

Questo particolare integrato rientra nella categoria degli oscillatori astabili ed è molto usato nella costruzione di generatori di segnali ad onda quadra che, però, non richiedano una particolare precisione e stabilità. I vantaggi che giocano a favore di questo integrato sono molteplici, primi fra tutti economia, versatilità e semplicità di impiego.

Le caratteristiche elettriche dell'integrato (alimentazione fra i 5 e 18 Volt; corrente di uscita 200 milliampere) lo rendono ideale nelle applicazioni in circuiti di semplice costruzione.

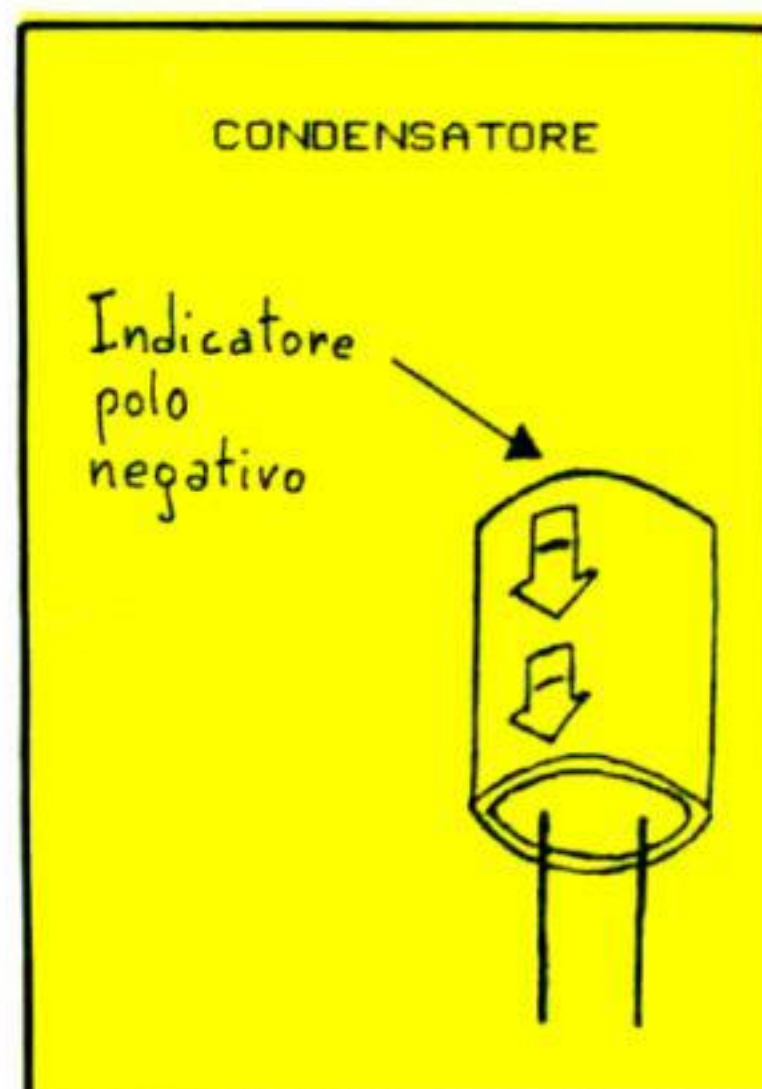
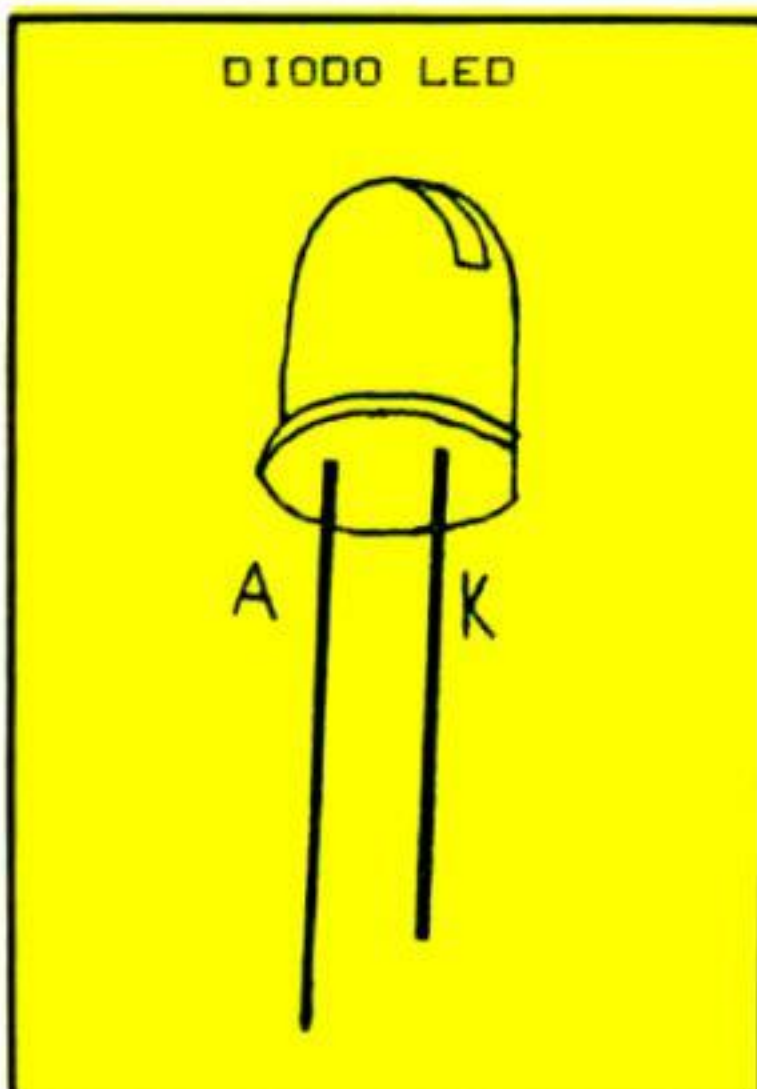
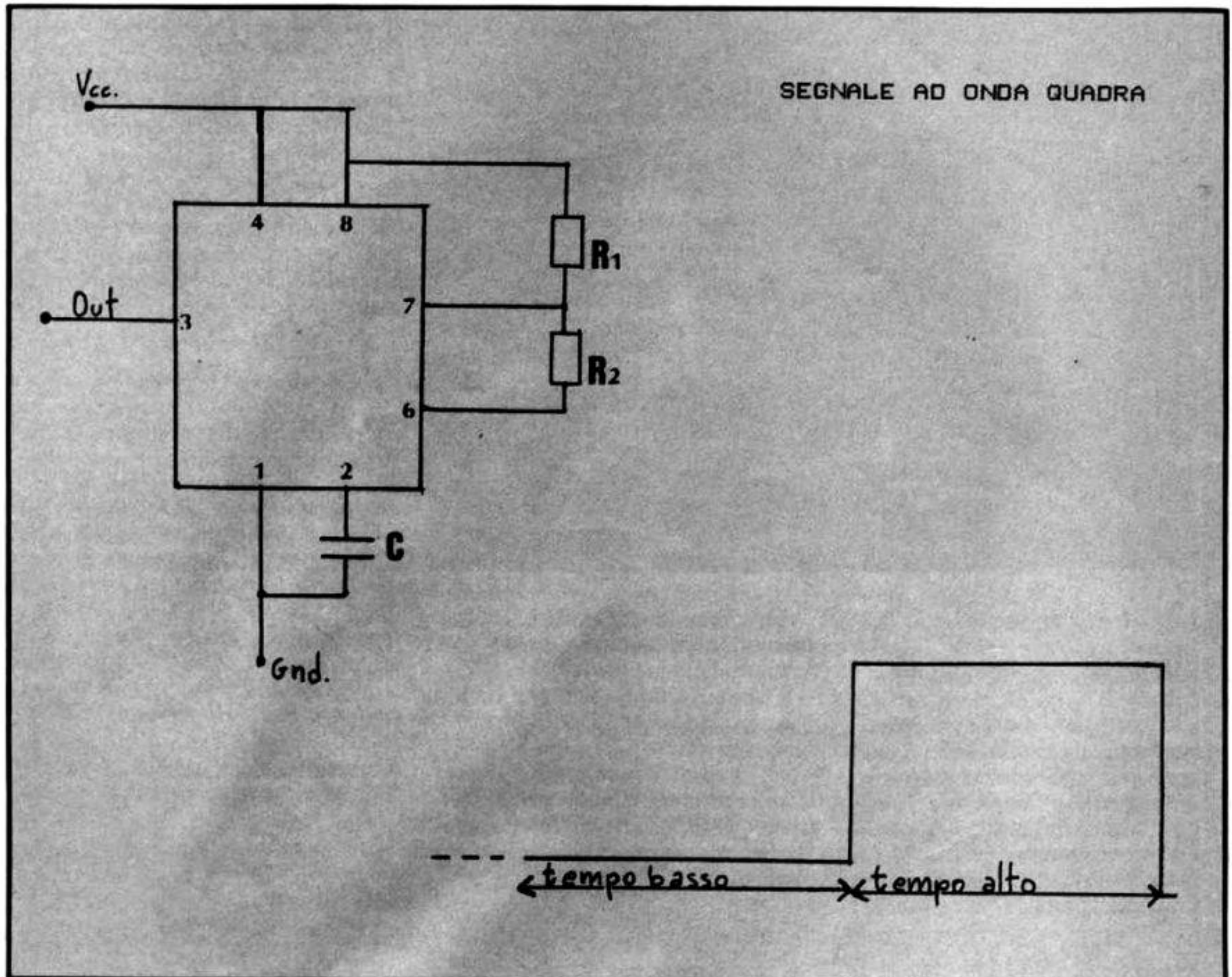
E' di vitale importanza rispettare i limiti di funzionamento sopraindicati; in caso contrario l'integrato potrebbe venir seriamente danneggiato.

Come funziona l'integrato:

Un'onda quadra è costituita da una successione regolare di impul si, alternativamente alti e bassi. La frequenza, cioè il numero di volte in cui gli impulsi si ripetono in un secondo, è determinata dalle seguenti leggi:

frequenza = 1.44/(R1+2*R2)*C
tempo alto = 0.693*(R1+R2)*C
tempo basso = 0.693*(R2)*C

Accorgimenti per il montaggio

Riteniamo sia molto utile seguire alcuni suggerimenti per la buona riuscita del montaggio dell'AUTOFIRE:

• Posizionare il supporto dell'integrato (zoccolo)

• Sistemare opportunamente (vedi schema circuitale) gli altri componenti

• Iniziare le operazioni di saldatura cercando di surriscaldare il meno possibile i componenti

• Inserire correttamente l'integrato nell'apposito supporto (la tacca di riferimento posta sul supporto deve corrispondere a quella dell'integrato)

• Prestare particolare attenzione all'inserimento del condensatore e del diodo led (vedi disegno)

Alimentazione

Il circuito viene alimentato da quattro pile (stilo) da 1.5 Volt collegate in serie per un totale di 6 Volt.

Tale soluzione si è resa necessaria perchè, purtroppo, i connettori dei joystick economici non possiedono, di solito, il pin relativo all'alimentazione disponibile sulla porta giochi. I lettori che non vogliano utilizzare le quattro pile "esterne", potranno ricorrere alla sostituzione del connettore di serie con uno completo, in modo da alimentare il circuito direttamente dal computer.

Collegare il polo positivo dell'alimentazione al morsetto positivo (+) indicato nello schema circuitale ed il polo negativo al morsetto negativo (-).

Collegamento del circuito al joystick

Terminato il montaggio del circuito, e verificata la correttezza dei collegamenti, è possibile procedere al collegamento dello stesso al joystick.

• collegare il morsetto negativo del circuito alla massa presente all'interno del joystick (la massa è facilmente

individuabile, osservando attentamente la scheda del joystyck si noterà un collegamento comune ai quattro interruttori delle direzioni)

• collegare il piedino 3 (uscita) dell'integrato NE-555 al filo proveniente dal pulsante di fuoco (non in comune agli altri interruttori)

• effettuare alcune prove di collaudo mediante il programma di queste pagine.

L'interruttore, naturalmente, servirà per disinnescare il circuito nel caso di videogame che non prevedano l'autofire.

Si consiglia di spegnere il circuito (o disconnettere il joy) durante il normale uso della tastiera: caratteri casuali potrebbero apparire alla pressione dei tasti.

SCHEDA TECNICA

Hardware e software per computer: C/64
Hardware per computer C/16 Plus/4 C/128 Vic-20

Software facilmente adattabile agli altri computer Commodore

Richiede joystick

Consigliato ai lettori che desiderino iniziare piccoli esperimenti di hardware

```
100 REM  COMMODORE 64
110 :
120 REM  AUTOFIRE TEST
130 :
140 PRINT:PRINT"RILOCAZIONE ? 6
    80";
150 FOR A=1 TO 5:PRINTCHR$(157)
    ;:NEXT
160 INPUT R
170 :
180 FOR A=0 TO 17:READ B:C=C+B
190 IF B=INT(B) AND B>=0 AND B<
    =255 THEN 210
200 PRINT"ERRORE NEI DATI";:END

210 POKE R+A,B:NEXT
220 IF C<>1879 THEN 200
230 :
240 PRINT
250 PRINT"PER TESTARE L'AUTOFIR
    E:"
260 PRINT
270 PRINT"1) SYS"R
280 PRINT"2) JOYSTICK IN PORTA
    2"
290 PRINT"3) AZIONARE AUTOFIRE"
300 PRINT"4) DISABILITARE AUTOF
    IRE"
310 PRINT"5) RUN/STOP RESTORE"
320 :
330 REM  INIZIO DATI
340 :
350 DATA  120,173,0,220,41,16,7
    4,141,29
360 DATA  212,74,74,74,141,32,2
    08,16,239
370 DATA  -1
380 :
390 REM  FINE DATI
```

```
100 REM  COMMODORE 64
110 :
120 REM  AUTOFIRE TEST
130 :
140 PRINT:PRINT"RILOCAZIONE ? 6
    80";
150 FOR A=1 TO 5:PRINTCHR$(157)
    ;:NEXT
160 INPUT R
170 :
180 FOR A=0 TO 16:READ B:C=C+B
190 IF B=INT(B) AND B>=0 AND B<
    =255 THEN 210
200 PRINT"ERRORE NEI DATI";:END

210 POKE R+A,B:NEXT
220 IF C<>2429 THEN 200
230 :
240 PRINT
250 PRINT"PER TESTARE L'AUTOFIR
    E:"
260 PRINT
270 PRINT"1) SYS"R
280 PRINT"2) JOYSTICK IN PORTA
    2"
290 PRINT"3) AZIONARE AUTOFIRE"
300 PRINT"4) DISABILITARE AUTOF
    IRE"
310 PRINT"5) RUN/STOP RESTORE"
320 :
330 REM  INIZIO DATI
340 :
350 DATA  120,169,16,24,44,0,22
    0,208,250
360 DATA  176,249,238,32,208,56
    ,176,243
370 DATA  -1
380 :
390 REM  FINE DATI
```

Passeggiando tra Peek, Poke e Sys

Un viaggio tra gli indirizzi di memoria del C/16 e del Plus/4

di Fabio Calabrò

SCHEDA TECNICA

Informazioni a scopo didattico di utilità generale.

Idonee per computer C/16 e Plus/4, non utilizzabili per altri computer Commodore.

Consigliato ai principianti

Considerato il successo della rubrica Peek Poke & Sys dedicata al C/64, ecco un intero articolo dedicato allo studio di particolari indirizzi di memoria, di vettori e di puntatori, nonchè di routine del Sistema Operativo dei due computer.

Reset parziale

La routine in linguaggio macchina che inizia dalla locazione $8000 serve per resettare parzialmente il computer poichè i colori e molte locazioni di memoria rimangono invariate. Provate a digitare, in modo diretto, la seguente linea:

Color 0,2,0: Color 4,2,0: Color 1,3,5

Tali comandi muteranno i colori dello sfondo, della cornice e del cursore; provate ora a scrivere, sempre in modo diretto, la seguente istruzione (si ricorda che 32768 è il decimale di $8000):

SYS 32768

Non appena premete il tasto Return (operazione che, in queste pagine, abbrevieremo con: [R]) il computer si resetterà solo parzialmente: i colori prima scelti, infatti, non sono cambiati.

Screen clear

La routine CINT, che fa parte delle routine del Kernal, serve per cancellare il video e colorare il cursore in nero. Tale routine, che corrisponde quindi al comando Basic "Scnclr" (a parte il cambiamento del colore), ha inizio a partire dalla locazione $FF81 (il cui decimale è 65409). La seguente riga, da digitare in modo diretto, ne evidenzia il funzionamento:

Color 1,3: For I=1 to 200: Print "Federica";: Next I: Sys 65409 [R]

Lo schermo dapprima di riempirà di scritte rosse ma in seguito (Sys 65409), richiamando la routine del Kernal, il colore del cursore tornerà nero e lo schermo si pulirà.

Auto Line

Le locazioni decimali 115 e 116 contengono il valore di incremento per il comando di numerazione automatica delle linee Basic in fase di digitazione.

Alterando il contenuto di tali indirizzi (da gestire nel solito modo byte basso / alto), pur senza impartire il comando auto line, lo si attiva automaticamente; il valore di incremento sarà rappresentato dal valore ottenuto dalla somma tra il contenuto del byte 116, moltiplicato per 256, ed il contenuto del byte 115; se tale somma è nulla, si annulla il modo auto line.

Provate a digitare in modo diretto la seguente riga:

Poke 115,8; Poke 116,0

Da questo momento il computer si comporterà come se fosse stato impartito il comando "Auto 8". Per verificare, digitate:

10 REM auto-on

Dopo [R] comparirà sul video il numero di riga 18, uno spazio ed il cursore lampeggiante, che indica che il computer attende la digitazione della line Basic numero 18.

Se si scrive in modo diretto, dopo aver premuto il tasto di reset, la riga...

Auto 100,10: Poke 115,0: Poke 116,0

...si esce dal modo di auto line dopo esservi appena entrati!

Tono e volume

Provate a digitare direttamente la seguente riga seguita ovviamente da [R]:

Vol8: Sound 3,23,20

Il computer emetterà un suono tramite la voce 3 per la durata di 20 sessantesimi di secondo; il numero 23 rappresenta il valore del tono e del volume del suono emesso; provate, infatti, a digitare la riga:

Print Peek(126) + 256 Peek(127)*

Sullo schermo, dopo [R], comparirà il numero 23. Questo perchè i byte 126 e 127 contengono il tono e il volume dell'ultimo comando Sound nella forma byte basso / alto.

Tasto Run/Stop

Le locazioni decimali 786 e 768 riguardano principalmente il tasto Run/Stop. Pokando in tali locazioni, rispettivamente, i valori 69 e 26, si disabiliterà il tasto in questione. Questa possibilità permette protezioni che evitano l'interruzione di un programma, anche nel caso in cui si ricorra alla pressione accidentale(?) del suddetto tasto.
Provate a digitare quanto segue:

Poke 786,69: Poke 768,26: Do: print "Pippo";: Loop [R]

Il ciclo infinito di stampa non potrà essere interrotto tramite il tasto Run/Stop in quanto è stato precedentemente disabilitato: provare per credere!

Se si ricorre alla pressione contemporanea di Run/Stop e del tasto Reset non sarà comunque possibile "rientrare" in Basic.
I valori "normali" dei due byte sono, rispettivamente, 66 e 134.

Colore principale

La locazione decimale 134 contiene sempre il colore del cursore. Tale locazione, però, è manipolata dal computer durante la colorazione e non ne tiene conto per l'effettiva modifica cromatica. Pokando un valore diverso, in altre parole, non si ottiene una modifica del colore.

Per i principianti

Queste pagine, ovviamente, sono adatte anche (e soprattutto) per i principianti; per gli esperti rappresentano, comunque, un valido strumento per sofisticare i propri programmi; è bene chiarire, fin da ora, i significati di alcuni termini che saranno molte volte usati.
La memoria del computer è divisa in byte indirizzabili (chiamati anche locazioni oppure celle); molti di questi indirizzi svolgono funzioni ben definite e, a seconda dal loro contenuto, il Sistema Operativo si comporta in un certo modo.
Il contenuto di alcuni byte posti in Ram (il cui valore è sempre compreso tra 0 e 255) può essere sia letto che modificato.
Non tutti i byte possono essere alterati (tutti, però, possono essere letti) perchè il Sistema Operativo li utilizza per funzioni particolarmente importanti e delicate: una modifica del loro contenuto potrebbe rendere inservibile il computer.
Il comando per modificare il contenuto di un byte è Poke; la sua sintassi è:

POKE X,Y

in cui X corrisponde alla cella di memoria di cui si vuole mutare il contenuto, ed Y corrisponde al valore (compreso tra 0 e 255) che si vuole immettere in quella stessa locazione.
Il comando che permette di leggere il contenuto di un byte è Peek, la cui sintassi è:

PRINT PEEK (X)

che stampa su video il contenuto della locazione X (numerata tra 0 e 65535).

Per "vettore" si intende la coppia di due byte consecutivi che indicano l'indirizzo di memoria da cui parte una certa routine in Linguaggio Macchina; (N.B: i vettori, in Basic, sono tutt'altra cosa).
Per esempio, la routine di Save è allocata a partire dall'indirizzo esadecimale $F1A4; ogni volta che si ordina al computer di salvare il programma in memoria, viene eseguita la routine che ha inizio dalla cella di memoria indicata dal vettore di Save (costituito dai byte 816-817). Ma com'è possibile che due byte indicano un solo indirizzo? Non essendo possibile memorizzare in un solo byte un numero maggiore di 255, si ricorre al sistema di byte basso / byte alto; in pratica la somma del byte basso e del prodotto del byte alto, moltiplicato per 256, corrisponde all'indirizzo di memoria a cui si vuole realmente "puntare".
Nel nostro esempio la routine di Save allocata a partire da $F1A4 (e quindi dal decimale 61860), viene richiamata dal vettore 816-817; con...

Print Peek (816) + 256 Peek (817)*

...si ottiene 61860.
E' intuitivo che, alterando il contenuto di un vettore, è possibile far eseguire una routine diversa da quella prevista.

Riferendoci sempre nel nostro esempio, se facciamo puntare il vettore 816-817 alla routine di LOAD invece che di SAVE, il risultato è sorprendente: ad ogni comando di salvataggio ne corrisponderebbe uno di... caricamento!

Il Sistema Operativo (S.O.) è costituito da numerosi programmi in Linguaggio Macchina (L.M.) che adempiono a funzioni vitali per il computer. I byte di cui si parlava prima, di cui non si può cambiare il contenuto, sono proprio quelli che formano le istruzioni del Sistema Operativo. Questo, infatti, è residente su memoria ROM (Read Only Memory); vale a dire una memoria che non può essere mutata dal programmatore, ma solo utilizzata.

L'istruzione Basic SYS (X) serve per far partire l'esecuzione di un programma in Linguaggio Macchina, locato a partire dalla locazione X.
Essendo, è bene ripeterlo, il S.O. un insieme di routine in Linguaggio Macchina non riprogrammabili, utilizzando l'istruzione Sys è possibile far eseguire direttamente un programma del Sistema Operativo.
Per esempio, a partire dalla locazione $FF81 (esadecimale di 65409) è memorizzata la routine che inizializza l'editor di schermo (in altre parole pulisce lo schermo). Impartendo, allora, il comando...

SYS 65409

...il computer esegue la routine locata a partire da $FF81 e pulisce, quindi, lo schermo.

Per cambiarne il contenuto si deve quindi ricorrere al solito comando "Color".
Provate a digitare la seguente riga:

Color 1,3,5: Print Peek (134)

Sullo schermo sarà stampata la somma della luminosità moltiplicata per sedici con il numero del colore diminuito di 1; in parole più semplici:

*(16*5)+(3-1)=82*

N.B: il nero ha come unica luminosità il valore 7.
Provate ora a scrivere:

Poke 134,10

Il colore del cursore non cambia.

Check-up del Kernal

L'indirizzo di memoria 144 corrisponde al byte di stato del Kernal. In altre parole leggendo il contenuto di questo byte è come se impartissimo il comando:

Print ST

Il valore di questo indirizzo dipende dal risultato dell'ultima operazione di Input/Output; in generale, se il contenuto è nullo, vuol dire che l'operazione di I/O è andata a buon termine.
Provate, per esempio, a eseguire le seguenti operazioni:

- *Spegnete il drive.*
- *Digitate: DLOAD "pippo" oppure un qualsiasi altro nome.*
- *Digitate in modo diretto, dopo l'inevitabile segnalazione di errore, quanto segue:*

Print ST: Print Peek (144)

Non essendo andata a buon termine (Device Not Present Error) l'ultima operazione di I/O, sullo schermo compariranno due 66; il primo è il contenuto della variabile riservata ST, il secondo è il contenuto del byte 144.

Flag di Rvs

Il byte 194 è il flag del modo di scrittura in Reverse.
Se si sta operando "normalmente" il valore è zero, mentre è 18 se si scrive in Reverse; questa cella la si può sia leggere sia scrivere e per cambiare modo di scrittura, senza ricorrere nè ai codici Ascii nè ai tasti di Reverse, si può agire su questo byte.
Provate a digitare, in modo diretto, quanto segue:

Poke 194,0: Print "PIPPO": Poke 194,18: Print "PLUTO": Print "MINNI" [R]

Osservate il risultato: il cambiamento del modo di scrittura tramite un pokaggio del byte 194 è valevole SOLO per la PRIMA stampa successiva al comando Poke, a patto che, ovviamente, non sia presente un carattere di punto e virgola (;) o una virgola (,).

Flag di pausa

Il byte 240 corrisponde al flag di pausa; se il contenuto è 128 il computer attende la pressione di un tasto prima di eseguire la successiva istruzione; una volta premuto un tasto qualsiasi, il contenuto del byte si annulla automaticamente.
In pratica, settando 128 nel byte 240, è come se impartissimo il comando GETKEYA$, con la differenza che quest'ultimo non si può usare in modo diretto.
Provate a digitare la seguente riga in modo diretto:

PRINT "premi un tasto": POKE 240,128: PRINT "ok" [R]

Il computer attenderà la pressione di qualsiasi tasto prima di continuare (e quindi prima di stampare "ok").

Nome file

Il byte posto alla locazione 622 indica la lunghezza del nome dell'ultimo file caricato, salvato o verificato.

Provate a eseguire quanto segue:

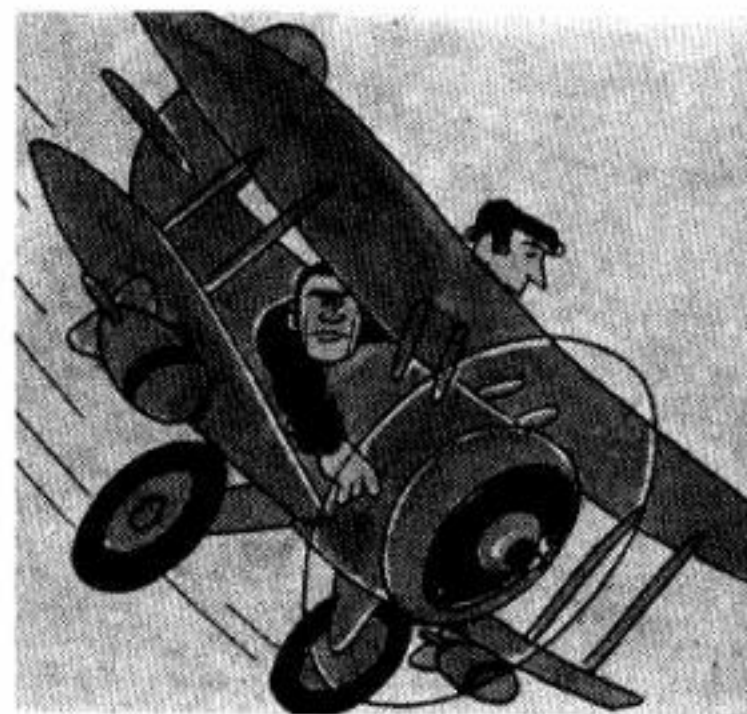
- *Spegnete il drive*
- *Resettate il computer (in modo da pulire la memoria)*
- *Digitate: 10 REM prova [R]*
- *Digitate, anche se è un errore: DSAVE "pippo" [R]*
- *Digitate: Print Peek(622)*

Sullo schermo comparirà il numero 5, poichè la lunghezza dell'ultimo file (in questo caso, salvato) è composto da 5 caratteri ("pippo").

L'area di memoria compresa tra 606 e 621 contiene i codici ASCII del nome del file; leggendo di seguito i byte a partire dalla locazione 606 per il numero di volte indicato dal byte 622 è possibile ottenere il nome del file.

A questo punto in memoria, dalla locazione 606 in poi, dovrebbe ancora essere il nome "pippo". Leggendo, quindi, i byte da 606 a 610 avremo a disposizione i codici ASCII dei 5 caratteri che compongono il nome.
Provate, infatti, a digitare, in modo diretto, la seguente riga:

FOR I=1 TO PEEK(622):PRINT CHR$ (PEEK(I+605));:NEXT I

Sullo schermo comparirà il nome "pippo". Ovviamente se, quando caricate o verificate un file, indicate il nome abbreviato con l'asterisco, dalla locazione 606 in poi sarà memorizzato il nome abbreviato compreso l'asterisco.

Scala

Nelle modalità multicolor e ad alta risoluzione, la variazione della scala degli assi si può cambiare con il comando SCALE.

Con il comando SCALE 1 si attiva la modalità di variazione e le coordinate, sia X che Y, vengono scalate da 0 a 1023 invece dei normali valori di scala che sono i seguenti:

- *Multicolor:*
  X=da 0 a 159
  Y=da 0 a 199
- *Alta Risoluzione:*
  X=da 0 a 319
  Y=da 0 a 199

Con il comando SCALE 0 si disattiva la modalità di variazione. Il modo scala (che assume solo i valori 0 oppure 1) viene memorizzato nel byte 742, che può essere sia letto sia scritto.

Si può, quindi, cambiare il modo SCALA, oltre che con il comando Basic "scale", anche agendo su questo byte, memorizzando uno zero (modo scala disattivato) o un uno (modo scala attivato).

Provate a digitare il seguente mini-programma e, ovviamente, alla fine impartite il Run:

10 GRAPHIC 1,1:POKE 742,0: DRAW 1,10,10 TO 80,80
20 POKE 742,1:DRAW 1,10,10 TO 80,80
30 GETKEY A$: GRAPHIC 0,0

Quando il computer incontra il comando Poke 742,0 è come se incontrasse il comando Scale 0; mentre, invece, quando incontra il comando Poke 742,1 si comporta come se avesse incontrato il comando Scale 1.

Osservate a questo punto il grafico ottenuto su video: noterete che, a parità di coordinate, le semirette ottenute sono posizionate diversamente; questo perchè la prima semiretta è stata stampata con i valori X e Y "normali" mentre la seconda è stata disegnata dopo avere scalato i valori di X e Y; si noti che in entrambi i casi i valori di X e Y sono eguali!

BANCA DATI
Al 32.70.226 dalle 13 alle 8 è in funzione la banca dati Modem-Shop Nuova Newel.

Novità drive compatibile 64/128 SLIM (TURBO) .... **L. 350.000**

Tutto per il tuo **Commodore AMIGA** oltre 400 **PROGRAMMI** software già disponibili.

Amiga PAL + 512K + Monitor colori + MOUSE .... **L. 1.850.000 + IVA**
Digi-View (digitalizzatore di immagini originale CBM) .... **L. 379.000 + IVA**
Drive 2 per AMIGA (nuovo modello «SLIME») .... **L. 349.000 + IVA**
Esp. di memoria 2 MEGA (novità)
L. telefonare

NOVITA' HARDWARE PER C-64/128
O.M.A. PLUS
RIVOLUZIONARIA, permette di trasferire IN UN UNICO FILE da disco a disco, da nastro a disco, da disco a nastro e da nastro a nastro il 99,99% del software protetto!!!
IN 3 MINUTI ESEGUE TUTTO IL LAVORO!!!
I programmi così trattati possono essere caricati con l'apposito turbo menù fornito nel disco lavoro. O.M.A. sostituisce così tutti gli sprotettori e i copiatori su cartuccia (e non). E l'ultima e definitiva cartuccia per eseguire tutte le tue duplicazioni, O.M.A. consente inoltre ai più esperti la manipolazione del programma allo scopo di modificarne gli sprite, la musica, il numero delle vite disponibili .... **L. 79.000**

EPROM 802
Favolosa, semplice da montare con istruzioni in italiano. Indispensabile EPROM che trasforma la tua 802 in 803, con la possibilità di usare tutti i programmi grafici come, KOALA, PRINT SHOP, GEOS V. 1 e 2 .... **L. 45.000**
MPS - 801 (NEW) .... **L. 35.000**

SPEED DOS
Il migliore e più collaudato velocizzatore, copia dal disco, anche protetto, in 21 secondi, legge i 202 blocchi in 10 secondi, tasti funzione, hardcopy, comandi al D.O.S. diretti .... **L. 49.000**

MODEM TOTAL TELECOMUNICATION 64/128
Diretto senza interfaccia con software + manuale in italiano .... **L. 118.000**

HACKER, FREEZ, FRAME DA **L. 49.000**
sprotettori universali di cassette e dischi.

VOICEMASTER
Incredibile, fai parlare/cantare il tuo computer: puoi digitalizzare la tua voce, le più belle canzoni e persino programmare l'apparecchio in modo che riconosca la tua voce e ti risponda. Sì! E' incredibile, ma è vero. In dotazione, oltre al Voicesystem, un disco software e un microfono. Veramente Eccezionale!!! **L.115.000**

NUOVA NEWEL sas
Attualità elettroniche e Microcomputers
20155 MILANO - Via Mac Mahon, 75
Tel.: neg. 02/32.34.92 - uff. 32.70.226

OLTRE 200 PROGRAMMI DISPONIBILE PER COMMODORE 128 & CP/M

IN ANTEPRIMA ASSOLUTA LA FAVOLOSA FINAL CARTRIDGE III 64/128

L'evoluzione continua!!!

Eccovi l'ultima versione della mitica cartuccia!!!
Turbo la favolosa routine dello speed-dos su cartuccia 10 volte più veloce, (non necessita di elaborazioni) esterne nè al computer nè al floppy).
- 8 tasti funzione programmati
- 24K ram extra per i programmi in Basic
- un Supertoolkit tipo O.M.A. incorporato (.... dischi e cassette in un solo file)
- inoltre incorporato il Game Killer (finita la collisione degli sprite, vite, infinite...)
- permette di fare l'hardcopy del video con un solo comando in 12 gradazioni
- sempre a disposizione oltre 40 comandi BASIC

Queste e molte altre funzioni nella **FINAL CARTRIDGE III a sole** .... **L. 99.000**

PER ULTERIORI INFORMAZIONI RICHIEDERE I CATALOGHI PER IL TUO COMPUTER INVIANDO L. 1.000 in francobolli

COMMODORE AMIGA, 64/128, C16, MSX, OLIVETTI PRODEST, SPECTRUM 16/48/128, SINCLAIR, QL, ATARI 500/140, PC, compatibili........

Testando la tastiera

Come costringere il C/64 ad eseguire i nostri ordini impartiti mediante i tasti funzione

di Vincenzo Degni

Premettiamo che il presente articolo tratta argomenti strettamente legati alla tecnica di manipolazione dell'interrupt; riassumeremo, tuttavia, che cos'è l'interrupt, nonostante tale argomento sia già stato più volte trattato su C.C.C.

La routine di interrupt

Ormai tutti i nostri lettori dovrebbero sapere che cosa sia l'interrupt.

Tale routine viene automaticamente attivata dal calcolatore ogni 60esimo di secondo: essa "interrompe" qualunque cosa il microprocessore stia eseguendo, ed effettua una lunga serie di controlli, neccessari al corretto funzionamento del computer.

Tra i numerosi controlli figura, tra i primi, quello sulla "scansione della tastiera", sul lampeggio del cursore e sull'aggiornamento delle due variabili del tempo (TI, TI$).

Naturalmente il computer è cosi veloce, nell' eseguire i controlli, che non ci accorgiamo minimamente della loro esecuzione.

La routine di interrupt è vettorizzata in RAM, nei puntatori 788 ($0314) e 789 ($0315); ciò significa che ogni 60esimo di secondo il calcolatore esegue la routine in linguaggio macchina il cui indirizzo iniziale è codificato nel modo seguente:

*Print Peek(788) + Peek (789)*256*

Gazie proprio a quest'ultima osservazione, nulla impedisce di modificare il contenuto delle due locazioni inserendo, al loro posto, l'indirizzo di una routine in L.M. da noi creata.

Agendo in tal modo, ogni volta che verrà generato un interrupt, il controllo passerà alla routine il cui indirizzo è stato modificato. Il risultato sarà un'esecuzione praticamente "continua" della nostra routine. Ricordiamo però che, al termine della routine, sarà indispensabile un salto alla originaria routine di interrupt ($EA31).

La scansione della tastiera

Come avevamo precedentemente affermato, fra i numerosi controlli dell'interrupt figura quello della "scansione della tastiera"; ogni 60esimo di secondo, quindi, il computer controlla il contenuto del vettore RAM 655-656 ($028F-$0290), che punta all'indirizzo 60232 del Kernal, chiamando la routine di "decodifica della tavola della tastiera" che permette al computer di ricevere ordini dalla tastiera stessa.

Considerando, però, che il puntatore della routine di scansione della tastiera è su RAM, possiamo modificare tale vettore e farlo puntare ad una nostra routine (naturalmente in L.M.).

Ad esempio potremmo "programmare" i tasti di funzione, in modo che alla pressione di uno di essi, vengano eseguiti particolari compiti.

Un accorgimento da non dimenticare assolutamente, è che al termine della routine WEDGE da noi creata, sarà comunque indispensabile un salto all'originaria routine di "scansione" della tastiera (60232).

Un esempio pratico

Il brevissimo programma pubbli-

cato in queste pagine è un banale esempio di ciò che si può fare: inserendo un WEDGE nella routine di "scansione" della tastiera, attivando tale routine e premendo il tasto F1, cambierà il colore del bordo; con il tasto F2 cambierà il colore dello sfondo, mentre premendo F3 cambierà il colore del cursore.

Per disattivare la routine è sufficente premere simultaneamente i tasti Run-Stop e Restore.

Può capitare, tentando di utilizzarlo con programmi che, a loro volta, modificano il puntatore della scansione della tastiera, che si verifichi un "crash" che costringa a spegnere e riaccendere il computer, rincominciando daccapo.

Gli esperti potranno esaminare il disassemblato commentato del programma dimostrativo per capire come realmente il programma funzioni.

Nota della Redazione

Vincenzo Degni, l'autore del presente articolo, è un ragazzino di dodici anni che ha terminato di frequentare la seconda media in una scuola di Cinisello Balsamo; lavora con caparbietà nonostante la modesta attrezzatura che si ritrova: un C/64, dotato di solo registratore, ed un libro sul linguaggio macchina che, incautamente, volemmo regalargli in occasione di una sua visita presso la nostra Redazione.

L'argomento trattato, come lo stesso Vincenzo ha ricordato, è già stato più volte affrontato su C.C.C. Queste pagine, quindi, vogliono, più che altro, rappresentare un "premio" per la costanza e l'impegno dimostrati dal giovanissimo collaboratore.

```
Disassemblato commentato
------------------------

C000 A9 00       LDA #$00  ; SPEGNE IL
C002 8D 0E DC    STA $DC0E; TIMER.
C005 A9 00       LDA #$00  ; METTE 0
C007 8D 8F       STA $028F; IN 655.
C00A A9 C0       LDA #$C0  ; METTE 192
C00C 8D 90 02    STA $0290; IN 656.
C00F A9 01       LDA #$01  ; RIACCENDE IL
C011 8D 0E DC    STA $DC0E; TIMER.
C014 A6 C5       LDX $C5   ; CARICA IN X IL
                           ; VALORE DELLA
                           ; LOCAZ. 197
C016 E0 04       CPX #$04  ; CONTINUA
                           ; SE IL TASTO
                           ; F1 E' PREMUTO
C018 D0 07       BNE $C021; NO,
                           ; SALTA IN C021
C01A AE 20 D0    LDX $D020; SI, ACC.53280
C01D CA          DEX       ; LO DECREMENTA
                           ; DI 1
C01E 8E 20 D0    STX $D020; SCRIVE IN 53280
C021 E0 05       CPX #$05  ; CONTROLLA SE
                           ; F3 E' PREMUTO
C023 D0 07       BNE $C02C; SE NO
                           ; VA IN C02C
C025 AE 21 D0    LDX $D021; SI, ACC. 53281
C028 CA          DEX       ; LO DECREMENTA
                           ; DI 1
C029 8E 21 D0    STX $D021; SCRIVE
                           ; IN 53281
C02C E0 06       CPX #$06  ; CONTROLLA SE
                           ; F3 E' PREMUTO
C02E D0 07       BNE $C037; NO, SALTA
                           ; IN C037
C030 AE 86 02    LDX $0286; SI.ACCUM.646
C033 CA          DEX       ; LO DECREMENTA
                           ; DI 1
C034 8E 86 02    STX $0286; SCRIVE IN 646
C037 4C 48 EB    JMP $EB48; SALTA A 60232
```

```
100 REM TESTANDO LA TASTIERA
110 REM DI VINCENZO DEGNI
120 :
130 K=49152:PC=0
140 FOR X=K TO K+57
150 READ A:PC=PC+A:POKE X,A:NEX
    T
160 IF PC<>7058 THEN PRINT"ERRO
    RE DI TRASCRIZIONE":END
170 SYSK
180 DATA 169,0,141,14,220,169,0
    ,141,143,2,169,192,141,144,
    2,169,1,141,14,220
190 DATA 166,197,224,4,208,7,17
    4,32,208,202
200 DATA 142,32,208,224,5,208,7
    ,174,33,208,202,142,33,208
210 DATA 224,6,208,7,174,134,2,
    202,142,134,2,76,72,235
220 END
```

Quattro nuove routine

Arricchite la raccolta di sottoprogrammi per personalizzare i vostri listati

A cura di Alessandro de Simone

17900 Gestione menu
(Commodore 64)

Si tratta di una semplice routine piuttosto modulare che permette la gestione di un certo numero di opzioni da inserire in un menu.

La scelta avviene, come in molti programmi professionali, tramite la visualizzazione in reverse delle varie opzioni, in funzione del movimento dei tasti cursore oppure del joystick.

Sarà sufficiente muoversi in alto e in basso, tramite i tasti cursore, e premere Return per confermare la scelta.

La gestione da tastiera può essere sostituita o integrata dall'uso del Joystick (in porta 2) che prevede la pressione del tasto Fire per la conferma.

In teoria è gestibile una lista di un numero qualsiasi di opzioni; nell'esempio dimostrativo ve ne sono soltanto 5, ma si tenga presente che un numero superiore a 11 crea problemi a causa dell'indispensabile preventivo DIMensionamento di un vettore che, come è noto, è "illegale" nelle routine della nostra enciclopedia.

I nomi delle opzioni sono contenuti nel vettore X0$ e il loro numero è associato alla variabile X0%; inutile dire che, per un corretto funzionamento, è necessario controllare che il numero delle opzioni corrisponda esattamente al valore di X0%

Al "ritorno" dalla subroutine la variabile X3 conterrà il numero relativo all'opzione desiderata e sarà così possibile, magari tramite un'istruzione On... Goto, accedere alla parte di programma indicata dall'opzione.

```
100 REM ESEMPIO D'USO
110 REM SCELTA MENU
120 REM IN REVERSE
130 REM CON CURSORE
140 REM O JOYSTICK
160 X0$(1)="PRIMA OPZIONE"
170 X0$(2)="SECONDA OPZIONE"
180 X0$(3)="TERZA OPZIONE"
190 X0$(4)="QUARTA OPZIONE"
200 X0$(5)="QUINTA OPZIONE"
210 X0%=5
215 PRINTCHR$(147)"A VOI LA SCE
    LTA..."
220 GOSUB 17900
230 PRINT:PRINT:PRINT"HAI SCELT
    O"X3
240 :
9999 END
17900 RE$(1)=CHR$(18):X1=1:POKE 1
      98,0:POKE 649,0
17910 PRINTCHR$(19):FOR X2=1 TO 7
      :PRINTCHR$(17);:NEXT
17920 FOR X3=1 TO X0%
17930 PRINTRE$(X3)X0$(X3)
17940 RE$(X3)=CHR$(146):NEXT
17950 POKE 649,10:GET X4$
17960 IF X4$=CHR$(17) OR PEEK(563
      20)=125 THEN X1=X1+1:IF X1>
      X0% THEN X1=1
17970 IF X4$=CHR$(145) OR PEEK(56
      320)=126 THEN X1=X1-1:IF X1
      <1 THEN X1=X0%
17980 IF X4$=CHR$(13) OR PEEK(563
      20)<112 THEN X3=X1:RETURN
17990 X3=X1:RE$(X3)=CHR$(18):GOTO
       17910
17999 REM  SCELTA MENU REVERSE
```

18000 Random programmabile
(Qualsiasi Commodore)

Ecco una semplicissima routine che permette di generare un numero intero Random compreso tra un minimo ed un massimo fissati a piacere dall'utente.
Il funzionamento è estremamente chiaro e non necessita di particolari spiegazioni.
Le variabili X0 e X1 contengono, rispettivamente, il valore minimo e quello massimo desiderati e, al ritorno, la variabile intera X2% contiene il numero casuale.
La straordinaria semplicità dell'algoritmo dovrebbe facilitare al lettore la stesura di un analogo sottoprogramma che consenta anche la generazione di numeri decimali.

```
  100 REM ESEMPIO D'USO
  110 REM GENERAZIONE NUMERO
  120 REM CASUALE INTERO
  130 REM TRA UN MINIMO E
  140 REM UN MASSIMO DEFINITI
  150 REM DALL'UTENTE
  160 :
  170 X0=-100:X1=112
  180 GOSUB 18000
  190 PRINTX2%:GOTO 170
  200 :
 9999 END
18000 X2%=X0+RND(.)*(X1-X0+.5)
18010 RETURN
18099 REM RANDOM PROGRAMX1BILE
```

18100 Paroliere
(Qualsiasi Commodore + 1541)

Si tratta di un'altra utility dedicata a chi fa uso di file sequenziali (Easy Script e simili).
Così com'è, la routine visualizza il contenuto di un file sequenziale, il cui nome è contenuto nella variabile X0$, sotto forma di singole parole considerando gli spazi e i caratteri di punteggiatura come separatori fra una parola e l'altra.
Si ritorna al programma principale quando il file termina oppure se si preme un tasto qualsiasi.

Se inserita in un programma più complesso, questa routine si presta alla costruzione di un vocabolario anche perchè considera solo i caratteri ASCII minuscoli e modifica quelli eventualmente maiuscoli, in modo da evitare doppioni; esempio: "ComPuTer" viene modificato in "computer".
Qualche consiglio per migliorare la routine:
- Creare un file oggetto che contenga tutte le parole lette da un altro file.
- Creare un "controllore" che scarti automaticamente le parole già presenti nel vocabolario.
- Aggiungere una routine di ordinamento alfabetico.

```
  100 REM ESEMPIO D'USO
  110 REM LEGGE FILE SEQUENZIALE
  120 REM E STAMPA LE PAROLE
  130 REM INTERE
  150 :
  160 INPUT  "FILE NAME ";X0$
  165 GOSUB 18100
  170 :
 9999 END
18100 X1$="":CLOSE 8:OPEN 8,8,8,X
      0$
18110 GET #8,X2$
18115 GET X3$:IF X3$<>"" THEN CLO
      SE 8:RETURN
18120 IF ST THEN CLOSE 8:RETURN
18130 IF X2$>=CHR$(193)  AND  X2$
      <=CHR$(218)  THEN  X2$=CHR$
      (ASC(X2$) AND 127)
18140 IF X2$>=CHR$(65)  AND  X2$<
      =CHR$(90)  THEN  X1$=X1$+X2
      $:GOTO 18110
18150 IF  X2$="'"  THEN  X1$=X1$+
      X2$
18160 IF (X2$=CHR$(32) OR X2$=CHR
      $(13) OR X2$=CHR$(39)) AND
      X1$<>"" THEN PRINTX1$:X1$="
      "
18170 GOTO 18110
18199 REM  PAROLIERE
```

18200 Fusi orari
(Qualsiasi Commodore)

Quante volte, dovendo telefonare allo zio d'America, vi siete chiesti: chissà che ore saranno laggiù?
Con questa breve routine sarete in grado di visualizzare, in funzione dell'ora corrente, le varie ore di altre sette città fra le più importanti del globo.
Nel vettore X0$ sono contenuti i nomi delle città stesse e la differenza in termini di ore dall'ora italiana.
In fase di elaborazione, tramite funzioni LEFT$ e RIGHT$ vengono visualizzate la stampa del nome della città e dell'ora corrente.
L'ora deve essere inserita in formato HHMM: se sono le 12 e 30 minuti il nostro input sarà 1230; ogni tentativo "illegale" come ad esempio un input con un numero di caratteri diverso da quattro oppure un'ora inesistente (es. 3421) o un input di caratteri alfabetici verrà intercettato ed emesso, di conseguenza, il messaggio ERR.

```
100 REM ESEMPIO D'USO
110 REM FUSI ORARI
120 REM VISUALIZZA ORA
130 REM CORRENTE DI
140 REM OTTO CITTA'
160 :
170 INPUT "ORE (ES. 1230) ";X1$
    :X9$=X1$
180 PRINTCHR$(147):GOSUB 18200:
    IF X0$="ERR" THEN PRINTX0$:
    END
190 TI$=X9$+"00"
200 PRINTCHR$(19) TAB(20)CHR$(1
    8)TI$:X1$=LEFT$(TI$,4)
210 PRINTCHR$(19):GOSUB 18200:G
    OTO 200
9999 END
18200 X0$(0)="ROMA          "
18205 X0$(1)="LONDRA      -1"
18210 X0$(2)="SOFIA       +1"
18215 X0$(3)="MOSCA       +2"
18220 X0$(4)="TOKYO       +8"
18225 X0$(5)="SYDNEY      +9"
18230 X0$(6)="LOS ANGELES -9"
18235 X0$(7)="NEW YORK    -6":X0$
      =""
18240 FOR X2=1 TO 4:X3$=MID$(X1$,
      X2,1):IF X3$<"0" OR X3$>"9"
       THEN X0$="ERR":RETURN
18245 NEXT
18250 IF LEN(X1$)<>4 THEN X0$="ER
      R":RETURN
18255 X4$=LEFT$(X1$,2):IF VAL(X4$
      )>24 THEN X0$="ERR":RETURN
18260 X5$=MID$(X1$,3,2):IF VAL(X5
      $)>59 THEN X0$="ERR":RETURN
18265 FOR X2=0 TO 7:PRINTLEFT$(X0
      $(X2),11);" ";
18270 X6=VAL(X4$)+VAL(RIGHT$(X0$(
      X2),2))
18275 IF X6>24 THEN X6=X6-24
18280 IF X6<0 THEN X6=24+X6
18285 PRINTRIGHT$(STR$(X6),2);":"
      ;X5$:NEXT:RETURN
18299 REM FUSI ORARI
```

Elenco delle ultime routine pubblicate

(Fra parentesi è riportato il numero di Commodore Computer Club su cui sono apparse)

Le routine di questo numero sono di Michele Maggi e Valerio Ferri

Compressore di testi

Un programma utile per risparmiare spazio su disco, ma che può esser sviluppato per gestire testi cifrati

di Armando Sforzi

Nell'ambito informatico una delle problematiche risulta essere la memorizzazione dei dati destinati alle più svariate elaborazioni.

Poichè è sempre presente la necessità di raccogliere grandi masse di informazioni, parallelamente ai supporti hardware si è sviluppato software dedicato alla compressione dei dati.

Con il termine "compressione" si intende un particolare processo di codifica dei dati affinchè siano memorizzati in uno spazio minore rispetto alle condizioni "normali". Naturalmente questa operazione deve essere reversibile: deve esistere un algoritmo, inverso al precedente, che riporti le informazioni allo stato primitivo per essere utilizzate.

E' evidente la comodità di simili utility che consentono di aumentare, virtualmente, la capacità delle memorie di massa. Purtroppo la progettazione di routine "comprimenti" non è cosa facile, essendo molto complicati, in genere, gli algoritmi che informano il processo di compressione e decompressione (ad esempio l'algoritmo di Huffman).

Alcune considerazioni sulla struttura del codice ASCII, normalmente utilizzato nella trasmissione dei dati, hanno, tuttavia, permesso la progettazione della routine presentata in queste pagine, che, correttamente applicata, permette una sensibile riduzione di spazio. Tale riduzione è dell'ordine del 25%, vale a dire che, ad esempio, un file lungo 40 Kbyte viene

allocato in 30 Kbyte, risparmiando circa 10.000 locazioni su supporto magnetico.

Esiste un solo limite: la routine di compressione si applica solamente ai codici ASCII compresi fra il 32 e il 94, oltre che al 13 (ritorno carrello).

Il range di valori indicato, limitato solo in apparenza, comprende tutti i caratteri che servono per scrivere testi: spazio, simboli matematici, numeri e lettere. Prima del codice 32 non c'è niente che possa realmente essere utile, escluso, naturalmente, il carattere 13, che nella routine di queste pagine viene codificato come una chiocciolina (64).

Dopo il carattere 94 vi sono solo segni semi-grafici che possono essere eventualmente sostituiti dai caratteri disponibili. Si faccia conto, insomma, di avere a disposizione i caratteri di una normale macchina per scrivere (con le lettere, però, o tutte maiuscole o tutte minuscole).

Come funziona il programma

Descriviamo le fasi della sezione comprimente:

- *Si controlla che i codici scritti nella zona di memoria da comprimere siano compresi fra 32 e 94. Il codice 13 viene accettato, ma sostituito, nella fase di compressione, con il codice 64.*

- *Segue il processo di compressione, seguito dai puntatori 251/252. Il processo termina quando viene incontrato il valore zero logico (cioè un byte con i bit tutti a 0).*

- *Si caricano 4 byte alla volta in un buffer posto dalla locazione 38 alla locazione 41.*

- *Si sottrae 31 a ciascun codice caricato.*

- *Si considera il quarto byte caricato, lo si spezza in tre parti e ciascuna di queste viene inserita, mediante operazione di OR, nei due bit alti (il 6 e il 7) dei tre byte precedenti.*

- *I tre byte, così trattati, vengono inviati sul bus seriale nel modo solito, con la routine Kernal di Chrout.*

- *Il processo ha termine quando si incontra un byte nullo.*

Riferiamoci, ora, alla routine che decomprime.

• *Si prelevano i segmenti superiori (il 6 e il 7) di ciascuno dei tre byte e, con questi si compone il quarto byte.*

• *Si effettua un'operazione di AND su ciascuno dei primi tre byte, per recuperarne il giusto valore.*

• *Si somma, in ciascuno dei 4 byte, il valore 31.*

• *Si scarica in memoria il contenuto, così ricostruito, dei 4 byte.*

Alla fine di questo processo avremo in memoria il file, nella sua struttura originaria, delimitato, al suo estremo superiore, da uno 0 logico.

La sintassi dell'istruzione è:

SYS XXX, in.mem, 0/1

in cui:

XXX rappresenta l'indirizzo di partenza della routine che, grazie al programma caricatore di cui è corredata, risulta essere completamente rilocabile.

In.mem. rappresenta il primo byte in cui è allocato il blocco di dati (da prelevare o da immettere).

Il terzo parametro è 0, se si vuole che la routine abbia effetto comprimente (trasferisce dalla memoria al drive); viceversa il parametro è 1 per l'operazione inversa di decompressione (da drive a memoria).

Il semplice listato dimostrativo dovrebbe risultare sufficientemente esplicativo.

E'da ricordare che la routine segnala un errore di Type Mismatch nel caso in cui incontri un carattere posto al di fuori dell'insieme legale (32/94).

SCHEDA TECNICA

Software applicativo per:

didattica
utilità
protezioni

Anche i programmi pubblicati in queste pagine sono contenuti nel disco "Directory" di questo mese.

Idoneo per computer: C/64

Non facilmente adattabile ad altri computer Commodore

Richiede il disk drive

Consigliato agli ex-principianti

```
0 LIST1-19:REM *** COLLECT 1.
  0 ***
1 REM COMPRESSORE DI DATI
2 REM PER CODICI ASCII COMPRE
  SI TRA
3 REM 32 E 94  INCLUSO IL 13
5 REM SINTASSI:
6 REM SYS XXXXX,(PRIMO BYTE),
  0
7 REM COMPRIME I DATI DA MEMO
  RIA A DRIVE
8 REM SYS XXXXX,(PRIMO BYTE),
  1
9 REM DECOMPRIME I DATI DA DR
  IVE IN MEMORIA
10 REM INDICARE IL PRIMO BYTE
   DELLA ZONA DI MEMORIA SU CU
   I OPERARE
11 REM IL FILE DEI DATI TERMIN
   A CON UNO 0 LOGICO
12 :
13 REM BY A. SFORZI 1987 C/64
15 :
19 REM ......RUN 20 PER INIZIA
   RE......
20 INPUT"INDIRIZZO DI LOCAZION
   E";I
22 X%=(I+182)/256:Y=(I+182)-X%
   *256
26 GOSUB30:IFK<>37454THENPRINT
   "ERRORE NEI DATA":END
29 POKEI+39,Y:POKEI+40,X%:END
30 READA:IFA=-1THENRETURN
35 POKEI+J,A:J=J+1:K=K+A:GOTO3
   0
40 REM ... INIZIALIZZA ...
50 DATA32,253,174,32,138,173,3
   2,247,183,132,251,132,253,1
   33,252,133,254
60 DATA32,253,174,32,138,173,3
   2,247,183
65 DATA192,0,240,11,164,251,20
   8,2,198,252,198,251
70 DATA76,000,000
80 DATA160,0,177,251,240,25
90 DATA201,13,240,8,201,32,144
   ,12,201,95,176,8,230,251,20
   8,236,230,252,208,232
100 DATA162,22,108,0,3,164,253,
    165,254,132,251,133,252
110 REM ... COMP ...
120 DATA160,0,162,0,177,251,240
```

```
    ,9,201,13,208,2
130 DATA169,64,56,233,31,153,38
    ,0,240,37,200,192,4,208,233
140 DATA160,0,165,41,41,3,162,6
    ,10,202,208,252,25,38,0
150 DATA153,38,0,200,192,3,240,
    8,70,41,70,41,192,0,208,227
    ,202
160 DATA134,253,166,184,32,201,
    255,160,3,185,37,0,32,210,2
    55,136,208,247
170 DATA166,253,224,0,240,13,16
    5,251,24,105,4,133,251
180 DATA144,164,230,252,208,160
    ,162,3,32,201,255,96
250 REM ... DECOMP ...
260 DATA169,0,133,254,166,184,3
    2,198,255,160,3,32,207,255
270 DATA153,37,0,201,0,208,2,23
    0,254,136,208,241
280 DATA160,0,185,38,0,41,192,5
    ,41,133,41,200,192,3,240,8
290 DATA70,41,70,41,192,0,208,2
    34,70,41,70,41,160,4
300 DATA185,37,0,201,0,240,11,4
    1,63,24,105,31,201,64,208,2
    ,169,13
310 DATA145,251,136
320 DATA208,233,165,254,208,13
330 DATA165,251,24,105,4,133,25
    1,144,173,230,252
340 DATA208,169,162,1,32,198,25
    5,96,-1
```

```
  0 REM DEMO COLLECT C64 1/C
  1 REM CREA UN FILE COMPRESSO.
  2 REM CARICARE E LANCIARE PRI
    MA
  3 REM LA ROUTINE COLLECT 1.0
  4 REM INDIRIZZO SUGGERITO:491
    52
  5 :
 10 POKE56,78:P-20000:INPUT"FIL
    E NAME";N$:Q$-N$+" COMPRESS
    O"
 19 REM CREA UN FILE
 20 FORJ-1TO4:READA$:L-LEN(A$):
    FORK-1TOL:X$-MID$(A$,K,1):P
    RINTX$;
 25 POKEP+U.ASC(X$):U-U+1:NEXTK
    ,J:POKEP+U,0:PRINT:PRINT
 30 FORJ-0TOU:PRINTPEEK(P+J)CHR
    $(PEEK(P+J));:NEXT
 39 REM CREA IL FILE NORMALE
 40 PRINT:PRINT:PRINT"[RVS]ORA
    SALVO IL FILE NORMALE"
 45 J=0:OPEN2,8,2,"0:"+N$+",S,W
    "
 50 PRINT#2,CHR$(PEEK(P+J));:IF
    PEEK(P+J)-0THENCLOSE2:GOTO1
    00
 60 J-J+1:GOTO50
 99 REM CREA IL FILE COMPRESSO
100 PRINT:PRINT:PRINT"ORA SALVO
     IL FILE COMPRESSO"
105 OPEN2,8,2,"0:"+Q$+",S,W":SY
    S49152,P,0:CLOSE2
110 END
120 :
130 DATA QUESTA FRASE RAPPRESEN
    TA UNA SEMPLICE PROVA NECES
    SARIA PER STABILIRE
140 DATA SE IL PROGRAMMA DI COM
    PRESSIONE FUNZIONA CORRETTA
    MENTE
150 DATA NATURALMENTE IL FILE H
    A SOLO UNA FUNZIONE DIMOSTR
    ATIVA E NON
160 DATA NECESSARIAMENTE IL RIS
    ULTATO OTTENUTO CONSENTE CO
    MPRESSIONI NOTEVOLI
```

```
  0 REM DEMO COLLECT 1.0
  1 REM DECOMPRIME UN FILE
  2 REM CARICARE E LANCIARE PRI
    MA
  3 REM LA ROUTINE COLLECT 1.0
  4 REM INDIRIZZO SUGGERITO:491
    52
  5 :
 10 POKE56,78:P-30000:J-0:INPUT
    "FILE NAME";N$:REM NOME DI
    UN FILE COMPRESSO
 20 OPEN2,8,2,"0:"+N$+",S,R":SY
    S49152,P,1:CLOSE2
 30 PRINTCHR$(PEEK(P+J));:IFPEE
    K(P+J)-0THEN50
 40 J-J+1:GOTO30
 50 END
```

Considerando che i numeri 1, 2 e 7 sono esauriti, vogliate inviarmi i numeri arretrati ....
al prezzo di L. 5.000 cadauno per richieste fino a 4 numeri. o di L. 4.000 cadauno per richieste oltre i 4 numeri arretrati, e perciò per un totale di L.............. Sono a conoscenza che i fascioli suddetti non saranno inviati in contrassegno e, pertanto. ho provveduto oggi stesso a versare il canone di L.................. a mezzo c/c postale n. 37952207 intestato a:
Systems Editoriale - V.le Famagosta, 75 - 20142 Milano

GIUDIZIO SUI PROGRAMMI DI QUESTO NUMERO

Ho assegnato un voto da 0 a 10 ai programmi che indico di seguito:

A/ .... Voto ....

B/ .... Voto ....

C/ .... Voto ....

D/ .... Voto ....

PICCOLI ANNUNCI..

....

....

....

....

....

CERCO/OFFRO CONSULENZA..

....

....

....

....

....

INVIARE IN BUSTA CHIUSA E AFFRANCANDO SECONDO LE TARIFFE VIGENTI A:

COMMODORE COMPUTER CLUB

V.le Famagosta, 75
20142 Milano

INVIARE TUTTA LA PAGINA ANCHE SE SI UTILIZZA UNA SOLA SCHEDA

Nome .... **Cognome** ....

Via .... **N°** .... **CAP.** .... **Città** ....

Telefono .... **Orario** ....

Quale fascicolo manca alla tua enciclopedia Commodore?

Per ordinare i fascicoli mancanti alla tua collezione di Commodore Computer Club utilizza l'apposita scheda in fondo alla rivista.

Viaggio all'interno del Basic

Come alterare (quasi) a piacimento i puntatori vitali dell'interprete Basic

(seconda e ultima parte)

di Alessandro de Simone

Secondo esperimento

Fase N.1

Spegnete e riaccendete il computer. Digitate, *così come è pubblicato* (senza alcuna modifica), il programma "Prova stringhe 1/a" privo della riga 180 in modo da evitare un "File not found error". Verificatelo e registratelo dopo aver aggiunto la riga 180.

```
100 rem prova stringhe 1/a
110 a$=" a. de simone "
120 b$="c.c.club"
130 c$="somma"+a$+b$
140 print a$,b$,c$: print
150 rem riga di riempitivo
160 rem riga di riempitivo
170 rem riga di riempitivo
180 load"stringhe n.1/b",8
```

Fase N.2

Spegnete e riaccendete il computer. Digitate e registrate (col nome "Stringhe n.1/b"), dopo un opportuno controllo, il listato dal nome corrispondente avendo l'accortezza di trascriverlo senza alcuna modifica. In particolare facciamo notare la *coppia di doppi punti* (::) presenti, dopo i REM, nelle righe 110 e 120.

```
100 rem prova stringhe 1/b
110 rem::"attenzione!!!!"
120 rem::"guarda!!"
130 print a$,b$,c$
```

Fase N.3

Spegnete e riaccendete il computer. Caricate e lanciate il programma "Prova stringhe 1/a" e lanciatelo: Sorpresa!

Invece di visualizzare, in seguito al caricamento automatico di riga 180, ciò che ingenuamente ci saremmo aspettati (le stringhe A$ e B$ di riga 110 e 120 del *primo* programma), compaiono sullo schermo i caratteri delle righe 110 e 120 del *secondo* programma nonostante siano preceduti da due istruzioni REM. Che cosa è successo?

Esaminiamo con attenzione il listato 1/A. Le righe 110 e 120 *definiscono* una stringa, mentre la 130 effettua una *concatenazione* tra stringhe. L'interprete, di conseguenza, individuerà A$ e B$ mediante puntatori che punteranno *all'interno* del programma. Per C$, invece, sia i puntatori, che i caratteri costituenti C$, si troveranno *al di fuori* dell'area Basic.

Quando la riga 180 carica il secondo listato, *tutti i puntatori*, come già visto nel primo esperimento, *non vengono modificati*, e se nel visualizzare C$ non sorgono problemi, ne sorgono, eccome, nel far apparire A$ e B$.

I puntatori di queste variabili, infatti, puntano ancora alle locazioni di inizio stringhe del *primo programma* che, dopo il caricamento del secondo listato, non contengono più i caratteri di riga 110 e 120, ma quelli situati dopo le REM del nuovo programma.

Ecco spiegato il perchè dell'insolita visualizzazione, come pure il motivo dei due caratteri di doppio punto situati dopo i REM: In questo modo i due programmi di figura 5 e 6 risultano (per ciò che riguarda le righe 100, 110 e 120) perfettamente identici agli effetti del funzionamento dei puntatori.

Conclusioni sul secondo esperimento

1) Il programma chiamante non deve contenere istruzioni del tipo: A$="NOME".

In caso contrario, se la variabile stringa viene interessata dal programma chiamato in overlay, nel migliore dei casi compaiono caratteri incomprensibili, nel peggiore possono verificarsi malfunzionamenti di tale entità da essere costretti ad interrompere l'elaborazione.

2) E' pertanto necessario, purtroppo, esaminare riga per riga il programma chiamante e modificare tutte le variabili stringa definite nel modo A$="nome". Un suggerimento: l'istruzione....

A$="NOME"
...modificatela in...
A$= ""+"NOME" (vale a dire stringa nulla + "NOME"
...oppure...
READ A$: A$=""+A$: DATA "NOME"

Terzo esperimento

Come caricare dapprima un programma breve e poi uno più lungo?

Alcuni testi suggeriscono di alterare i puntatori di inizio e fine Basic prima di effettuare l'operazione che, in effetti, è "proibita", dato che, come abbiamo visto, distrugge parte delle variabili già elaborate.

E' però possibile effettuare l'operazione in modo semplicissimo: Supponiamo di voler utilizzare dapprima il programma "Breve" che, giunto alla riga 120, riceve l'ordine di caricare il programma "Lungo" che occupa un numero di byte decisamente superiore.

Ebbene, le fasi da seguire sono le seguenti:

1) Digitare il programma più breve e registrarlo col nome "Breve".

```
100 rem programma breve
110 a=10: b=20: c=30
120 load"lungo",8
```

2) Digitare l'altro programma e registrarlo col nome "Lungo".

```
100 rem programma lungo
110 :
120 print a: print b: print c
130 :
140 rem queste righe
150 rem servono solo per
160 rem allungare il
170 rem listato
180 :
190 end: rem penultima riga
200 eseguire : run 210
210 load"breve",8
```

3) A questo punto (avendo cioè "Lungo" in memoria) digitare RUN 210. Questa operazione caricherà il programma "Breve" e lo renderà, come abbiamo avuto modo di studiare, di lunghezza eguale a "Lungo". Dopo l'elaborazione verrà richiamato (riga 120) nuovamente il listato di figura 8 che risulta essere di lunghezza pari (ma guarda un po'...) al programma chiamante!

I puntatori di inizio e fine Basic

Alcuni (presunti) malfunzionamenti del programma "Append" pubblicato sul N.42 di C.C.C. (pagina 5) offrono il pretesto per parlare ancora di quattro locazioni fondamentali per il corretto funzionamento del sistema operativo.
Allo scopo di meglio comprendere gli argomenti trattati nel presente inserto, si suggerisce al lettore di rileggere attentamente quanto pubblicato nell'inserto dello scorso numero.

Introduzione

I computer Vic 20, C-16, Commodore 64 allocano i programmi Basic, digitati da tastiera o caricati da unità magnetiche, a partire dall'indirizzo contenuto nelle due locazioni di memoria 43 e 44.
Riferendoci al Commodore 64, in tali locazioni sono normalmente contenuti i valori "1" (in 43) e "8" (in 44). L'indirizzo (I) della prima locazione in cui verrà allocato il programma Basic sarà dato dal semplice calcolo:

*I=1+8*256=2049*

...oppure da un più immediato...

*Print Peek(43)+ Peek(44)*256*

Per far funzionare correttamente un programma Basic è però necessario che il byte precedente tale locazione sia posto a zero; all'annullamento di tale byte provvede automaticamente il computer al momento dell'accensione.
Ne potete avere facilmente conferma chiedendo, mediante Peek, il valore ivi contenuto.
Provando, infatti, a digitare:

*Poke Peek(43)+Peek(44)*256-1,1*

Oppure, più semplicemente (ma è valido, ovviamente, per il solo C-64):

Poke 2048,1

noterete che alcuni comandi (NEW, RUN e altri) non vengono riconosciuti e viene emessa la segnalazione Syntax Error. Deriva, pertanto, la...

Prima regola

La locazione precedente un programma Basic deve SEMPRE contenere il valore nullo; qualsiasi altro valore provoca infatti inconvenienti di varia natura.
Inoltre...

Seconda regola

L'indirizzo della locazione di inizio Basic è data dal prodotto del contenuto di 43 sommato al contenuto di 44 moltiplicato per 256.
Omettendo tale verifica, di cui si è ampiamente parlato nell'inserto precedente, si possono verificare guai di vario tipo, per cui insistiamo nel ricordare la...

Terza regola

Ogni linea Basic è formata da:

• due byte di Link (=puntatori al prossimo byte di memoria Ram che rappresenta l'inizio della successiva linea Basic).

• due byte che rappresentano la numerazione della riga Basic stessa.

• un gruppo di (al massimo) 80 caratteri costituenti istruzioni, comandi e dati della linea Basic.

• un ultimo byte nullo. L'indirizzo di questo risulta quindi inferiore di un'unità al valore risultante dal calcolo dei puntatori di Link.
Se, ad esempio, la coppia di Link dovesse puntare alla locazione 10321, il byte nullo sarà rintracciabile all'indirizzo 10320.

Quarta regola

L'ultima linea di un programma Basic termina, invece che con un byte nullo, con tre byte nulli, in successione.

Quinta regola

L'indirizzo (I) dell'ultima locazione riservata al programma Basic è anche data dal calcolo dei puntatori 45 e 46:

*I= Peek(45)+ Peek(46)*256-1*

Tale indirizzo è l'ultimo byte nullo (dei tre) di cui si è parlato poc'anzi.
Le didascalie di figura 1 dovrebbero eliminare ogni incertezza.

Uno strano doppione

Da ciò che abbiamo detto sembrerebbe che il calcolatore dispone di due "sistemi" per individuare la fine di un programma:

- Indirizzo del byte puntato da 45 e 46.
- Tre byte nulli successivi.

In realtà l'interprete Basic del computer utilizza il primo sistema in alcuni casi ed il secondo in altri.
Quando il programma viene normalmente eseguito (e, in particolare, vengono utilizzate le istruzioni RUN, GOTO, GOSUB, ed altre), il calcolatore provvede a rintracciare la linea giusta iniziando a spulciare l'intero programma a partire dalla linea puntata da 43 e 44 e FERMANDOSI o nel caso in cui incontra l'indirizzo cercato (ed eseguendo le istruzioni ivi contenute), oppure nel caso in cui individua tre byte nulli in successione (ed emettendo, se del caso, un "Undefinèd Statement error").
Se, invece, si registra un programma su supporto magnetico, il sistema operativo provvederà a trasferire su disco (o nastro) il contenuto di tutti i byte compresi tra i due indirizzi puntati da 43 e 44 (=inizio) e 45 e 46 (=fine).

Le didascalie della figura 2, ed i due programmi pubblicati, chiariscono meglio quanto esposto.

L'alterazione dei quattro puntatori (43, 44, 45, 46) è la tecnica normalmente adoperata per il caricamento ed il salvataggio di programmi in Linguaggio Macchina.

Un commento a parte meritano i programmi pubblicati, scritti per un Commodore 64:

```
100 a=fre(0)
105 if a<0 then a=2    16-abs(a)
110 print a: end
120 poke 46,8: clr: goto100
130 poke 46,9: clr: goto100
140 :
150 rem programma n.1
```

Il primo, decisamente breve, occupa un numero di byte che è possibile calcolare in modo indiretto mediante FRE(0).

Le prime tre righe non fanno altro che calcolare, e visualizzare, il numero di byte RAM rimasti liberi.

Lo stesso risultato si ottiene digitando Run 120.

Con Run 130, invece, l'alterazione "forzata" del byte 46 sottrae artificialmente memoria alla RAM ed il risultato è diverso.

E' ovvio che il programma sembrerà più lungo del precedente, ma solo nel caso di registrazione (e successivo caricamento); al limite sarà possibile ottenere un messaggio di "Out Of Memory Error" nel caso si tentasse di digitare altre righe.

Se, infatti, modificate la riga 130 come segue...

```
130 poke 46,159: clr: goto100
```

...e tentate di aggiungere altre righe Basic (anche se "piene" di semplici Rem), come ad esempio....

```
200 rem aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
aaaaaaaaaa

210 rem bbbbbbbbbb bbbbbbbbbb bbbbbbbbbb bbbbbbbbbb
bbbbbbbbbb
```

...vi accorgerete molto presto di non riuscire ad aggiungere altre righe.

Se poi, addirittura, modificate come segue...

130 poke 46,160: clr: goto100

...otterrete subito un Out of memory con un semplice Run 120 ed il motivo è semplice da spiegare: il prodotto 160*256 indica una locazione il cui indirizzo (40960) è situato oltre quello consentito dal Basic.

Nel caso in cui non si sia verificato un Out of Memory, invece, la fine del programma verrà sempre individuata dall'interprete, solo dalla presenza di tre byte nulli che sono ancora al loro posto e che non sono stati alterati minimamemte dalle operazioni descritte in precedenza.

Il secondo listato altera uno dei due puntatori (i lettori più pignoli possono alterare anche quello contenuto nella locazione 45, in modo da puntare esattamente all'indirizzo desiderato).

```
100 print fre(0):rem verifica memoria prima
110 poke 46,15: clr: restore
120 print fre(0):rem verifica dopo
130 get a$: ifa$= "" then 130
140 f$="commodore computer club":rem 23 caratteri
150 for i=1 to 23
160 poke 15*256+i, asc(mid$(f$,i,1)): next
170 print chr$(14):rem maiuscolo minuscolo
180 clr: restore: print chr$(147)
190 for i=1 to 23
200 poke 1024+i,peek(15*256+i): next
210 :
220 rem   programma n.2
```

Il listato agisce in modo che il puntatore di fine Basic punti OLTRE il necessario e deposita, in questa zona, i caratteri della stringa F$ (cioè: "Commodore Computer Club"). In tal modo sarà possibile avere a disposizione una zona di memoria *inattaccabile* dal Basic in cui potrete trascrivere (ricorrendo al semplice algoritmo di righe 110-120) un messaggio di qualsiasi lunghezza.

Per rendervi conto del suo funzionamento, trascrivete il programma N.2, così come è pubblicato, e registratelo dopo averlo fatto girare almeno una volta.

Spegnete pure il computer, riaccendetelo e ricaricate il programma. Digitate RUN 170. Vi accorgete che il messaggio viene egualmente visualizzato (a dispetto di Run e CLR che dovrebbero cancellare ogni variabile senza pietà), dal momento che è rimasto "incorporato", al momento della registrazione, nel programma stesso!

Tale tecnica viene normalmente adoperata per salvare, insieme al programma Basic, brevi listati in linguaggio macchina, oppure il nome degli autori dei programmi (in modo da renderli "indelebili") o per tecniche di protezione.

Malfunzionamenti di append

Da quanto esposto dovrebbe risultare chiaro che la tecnica di Append, descritta nel N.42, non fa altro che seguire la procedura descritta nelle didascalie di figura 3.

Come mai, dunque, in alcuni casi il programma non funziona?

E' presto detto: sono molto diffusi, tra gli appassionati di computer, particolari utility (renumber, delete ed altre) il cui funzionamento non soddisfa pienamente la quarta e quinta regola precedentemente esposte.

In altre parole tali utility, al termine del loro lavoro, non fanno più coincidere l'indirizzo del byte successivo ai tre byte nulli (di fine programma) con l'indirizzo dei puntatori 45 e 46.

In genere, con tali utility, i puntatori 45 e 46 puntano qualche byte più "in là" dell'ultimo byte nullo col risultato, disastroso, di alterare completamente il sistema di linkaggio del programma Slave caricato in Append.

Il programma 3 risulta utile per individuare i programmi che possono generare malfunzionamenti in fase di Append.

```
63994 a1=peek(43) +peek(44)*256: a2=peek(45) +peek(46)*256
63995 for i=a1 to a2:if peek(i)=0 and peek(i+1) =0 and
peek(i+2)=0 then 63997
63996 next
63997 i1=(i+3)/256: i2=int(i1): i1=(i1-i2)*256
63998 print chr$(147)"poke45,"i1": poke 46,"i2": clr: restore"
63999 rem  programma n.3
```

Come si può notare, il listato è numerato con gli ultimi numeri disponibili con un C/64: tentando, infatti, di digitare...

64000 Rem

...si ottiene un Syntax Error.

Tale numerazione consentirà al lettore di inserire il listato stesso "in coda" ad un qualsiasi listato Basic, a patto, ovviamente, che non possegga eguali numeri di linea.

In conclusione:

• Il programma Append funziona correttamente a patto che almeno il programma Master sia "ortodosso". Questo, in altre parole, deve avere coincidenti i puntatori di fine Basic (45 e 46) con l'indirizzo successivo all'ultimo dei tre byte nulli.

• Nel caso in cui la fusione non avvenga, vuol dire che il programma Master non è ortodosso. In tal caso:

• Digitare il breve listato N.3 di queste pagine (o caricarlo da nastro o disco), e visualizzatelo con un semplice List.

• Mentre tale programma è ancora visualizzato sullo schermo, caricate il programma Master non ortodosso facendo in modo, ovviamente, che lo schermo contenga ancora il breve listato N.3.

• Terminato il caricamento risalite col cursore (senza cancellare lo schermo!) e premete il tasto Return su ciascuna linea Basic del programma 3: in tal modo esso verrà incolonnato in fondo al Master.

• Fate partire il programma con RUN 63994. Dopo un po' di tempo, proporzionale alla lunghezza del programma (a volte parecchie decine di secondi), verrà visualizzato un rigo contenente quattro istruzioni (vedi riga 63998). Posizionatevi sopra con il cursore e premete il tasto Return: in questo modo i puntatori 45 e 46 punteranno come di dovere.

• Registrate il programma così modificato, magari dopo aver cancellato le ultime righe (che costituiscono il programma N.3).

• Il programma, così registrato, è ora utilizzabile come programma Master in fusioni di append.

100 Rem append
130 Rem per C/64 e drive Commodore
140 print chr$(147)"nome prog.master": input x$

```
150 if len(x$)>16 then 140
160 print: print: print "nome prog.slave": input y$
170 if len(y$)>16 then 160
180 printchr$(147): fori=1to19: print: next
190 print "poke43," peek(43) chr$(157)":poke44," peek(44);
200 printchr$(19)"load" chr$(34) x$ chr$(34) ",8"
210 printchr$(19): fori=1to4: print: next
220 print "a=peek(43): b=peek(44):";
230 print "c=256*peek(46) +peek(45)-2": print: print
240 print "poke43,c and 255: poke44,int(c/256): new"
250 print: print: rem alessandro de simone
260 print "load" chr$(34) y$ chr$(34) ",8"
270 poke 198,10
280 for i=1 to 10: read a: poke 630+i,a: next
290 data 19,13,13,13,13,13,13,13,13,13,13
```

Figura 1

```
43/44   1/8   Puntatori di inizio
                (1+8*256=2049)

45/46   27/8  Puntatori di fine
                (27+8*256=2075)

2049     9       Primo link
2050     8       di linea
                 9+8*256=2057

2051   232       Numerazione
2052     3       prima linea
                 232+3-256-1000

2053   143       Rem
2054    32       spazio
2055    65       Carattere "a"
2056     0       Fine linea

2057    17       Secondo link
2058     8       di linea
                 17+8*256=2065

2059   242       Numerazione
2060     3       seconda linea
                 242+3*256=1010

2061   143       Rem
2062    32       Spazio
2063    66       Carattere "b"
2064     0       Fine linea

2065    25       Terzo link
2066     8       di linea
                 25+8*256=2073

2067   252       Numerazione
2068     3       terza linea
                 252+3*256-1020

2069   143       Rem
2070    32       Spazio
2071    67       Carattere "c"

2072     0       Fine
2073     0       area
2074     0       del Basic
```

Zona Basic coincidente con zona Load e Save: A mano a mano che vengono digitate le linee Basic, i due puntatori 45 e 46 vengono automaticamente aggiornati. Analogamente una routine provvede ad incorporare nella RAM la nuova linea Basic e ad inserire tre zeri, in successione, in coda ad essa (a patto che sia l'ultima, come numerazione). Si noti come il numero di linea Basic, qualunque esso sia, occupa sempre due byte come, pure i due byte di link. Ogni linea Basic termina sempre con uno zero. Nella figura, per motivi di semplicità, è rappresentato il banale listato...

1000 Rem a
1010 Rem b
1020 Rem c

...nel caso sia digitato su un C/64.

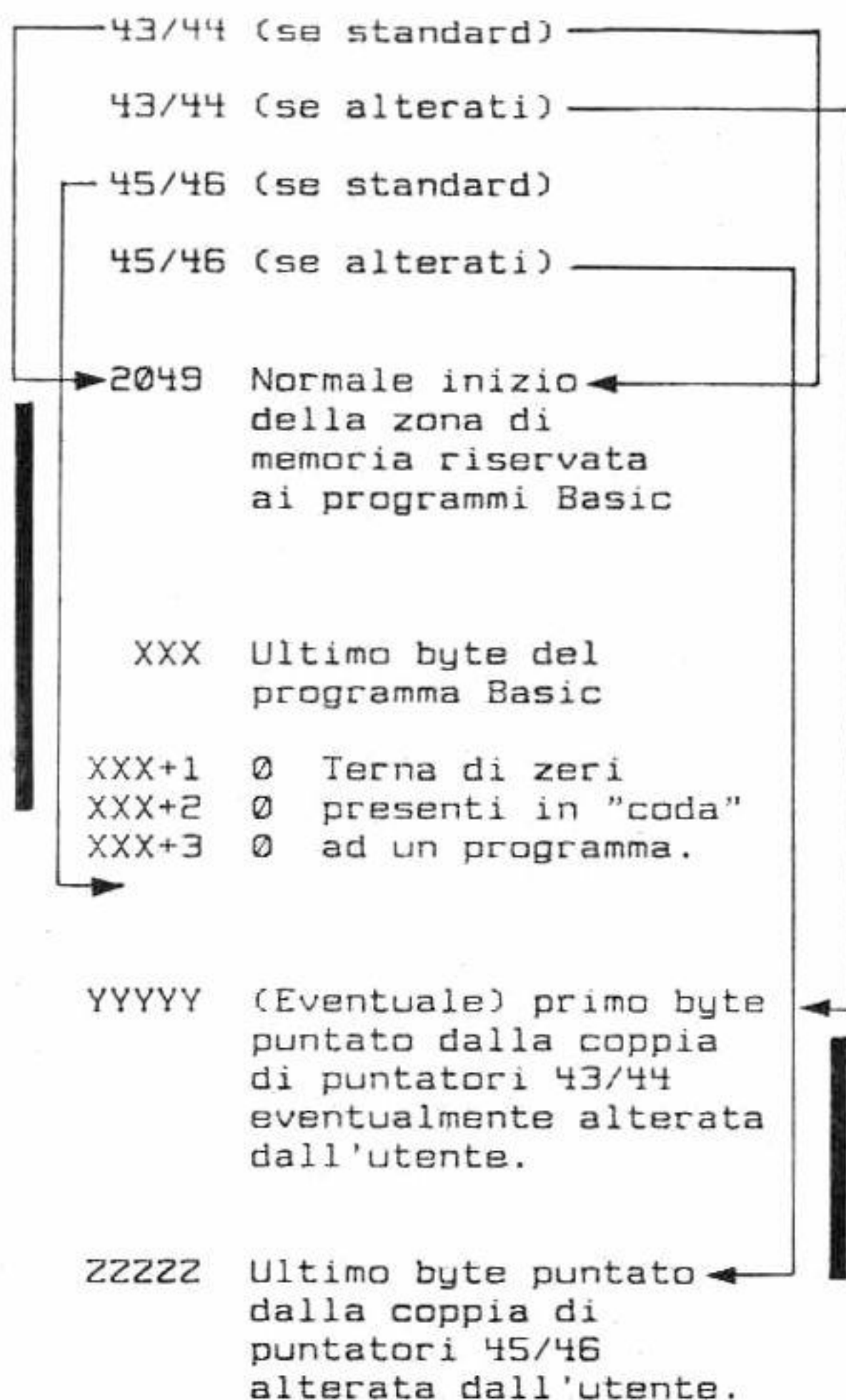
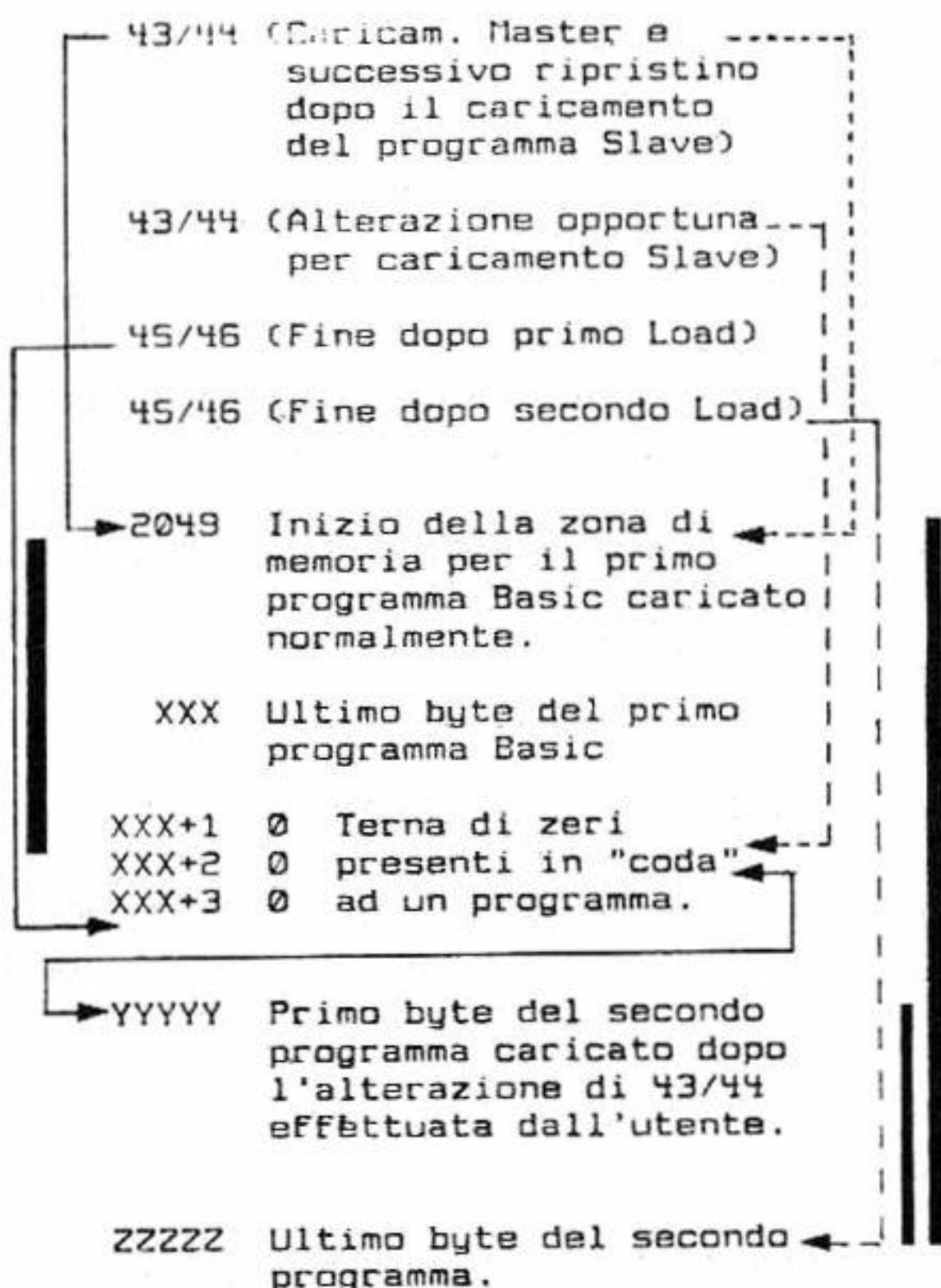
Figura 2

Manipolando opportunamente i byte 45 e 46 (=fine Basic) è possibile alterare le informazioni relative alla zona RAM interessata da Load e Save che, in caso contrario, coinciderebbe con la zona contenente il solo programma Basic. Nel caso illustrato in figura è possibile salvare (registrare) mediante Save, non solo il programma Basic ma anche la zona di RAM presente dopo di esso che può contenere eventualmente messaggi, variabili e, addirittura, programmi L.M. o sistemi di protezione.

Figura 3

Funzionamento dell'utility Append:
Fase 1: Accensione del computer e caricamento del programma Master. L'aggiornamento dei puntatori di fine Basic è automatico (frecce scure).
Fase 2: L'utente, altera i puntatori di inizio Basic in modo che "puntino" allo stesso indirizzo di fine Basic (45-46) meno due byte (a causa dei tre zeri successivi). (freccia tratteggiata in basso)
Fase 3: Il caricamento, automatico, "appende" in coda al Master, il programma desiderato (freccia tratteggiata in alto).
Fase 4: Terminato il caricamento l'utente ripristina i precedenti valori di inizio Basic (freccia punteggiata).

L'intera procedura è valida solo nel caso in cui il programma Basic sia ortodosso, simile, in altre parole, a quello di figura 1 e non di figura 2.

Oltre i puntatori

Vedremo, ora, di sfruttare la possibilità di alterare a piacimento i puntatori del Basic per realizzare insolite procedure.

Tra le varie applicazioni esamineremo in dettaglio la coesistenza di programmi Basic, l'utilizzo dell'area di schermo per allocare programmi, e la conseguente estrapolazione di tecniche insolite di protezione.

Su alcuni aspetti ci soffermeremo più a lungo di altri, anche per non privare i nostri lettori del piacere (!) di scoprire, da soli, altre possibilità.

Verrà ora descritta una tecnica, utile in più di un caso, che consente di "segmentare" a piacere la memoria Basic a disposizione dell'utente.

La segmentazione dell'area Basic

Abbiamo, dunque, imparato che un programma Basic, per il computer, è allocato a partire dalla locazione data da...

*Peek(43)+ Peek(44)*256*

...fino alla...

*Peek(45)+ Peek(46)*256*

...a patto, come ricorderete, che la locazione precedente l'inizio contenga un valore nullo.

Di ciò possiamo chiedere conferma con un semplice:

Print Peek(43); Peek(44)

che risponde con 1 e 8 (caso di un C/64 all'accensione).

Alterando i puntatori con...

*Poke 44,10: Poke (10*256+1)-1,0: New*

"costringiamo" l'interprete Basic a riconoscere, come spostata in avanti, l'area destinata ai programmi; in altre parole la prima locazione destinata al Basic sarà posizionata non più a

partire da 2049 (dato da 8*256+1), ma da 2561 (=10*256+1).

Con tale sistema abbiamo, come si dice in gergo, "spostato avanti" il Basic con la conseguenza di riservare 512 byte (2561-2049) per i nostri scopi più vari.

Potremo, tra l'altro, allocare routine in linguaggio macchina, (mezze) schermate, messaggi e, perchè no?, un altro programma Basic. Vedremo, una per una, le varie applicazioni.

Memorizza messaggi

Abbiamo già visto come "incorporare" un messaggio "in coda" ad un programma Basic; la tecnica descritta, però, impedisce un'eventuale, successiva, aggiunta di righe Basic, o la loro modifica, senza modificare anche il messaggio stesso, se non ricorrendo a complicate tecniche di puntamento.

Vi consigliamo di seguire alla lettera, come al solito, le note che seguono: è l'unico modo di non incorrere in errori:

- spegnete e riaccendete il C/64
- digitate:

Poke 44,10: Poke 2560,0: New

In questo modo sposterete l'inizio del Basic, come già detto, dall'originario 2049 a 2561.

- digitate il seguente programma:

```
100 print "digita un messaggio"
110 input a$: x=len(a$)
120 poke 2050,x
130 for i=1 to x
140 y$=mid$(a$,i,1)
150 poke 2050+i, asc(y$)
160 next
170 :
180 for i=1 to peek(2050)
190 y$ = chr$(peek(2050+i))
200 print y$;: next
```

Commentiamolo brevemente:
- Le righe 100/160 allocano da 2051 in poi, uno per uno, i caratteri che costituiscono la stringa A$, da digitare al momento dell'Input (riga 110). Si noti che la locazione 2050 servirà, in seguito, per individuare la lunghezza del messaggio stesso.
- Le righe 180/200 operano in modo inverso: estraggono dalla memoria Ram i valori precedentemente allocati, e ne stampano i codici Ascii.

• Verificato il corretto funzionamento (la stringa digitata viene visualizzata sullo schermo), digitate quanto segue:

Poke 44,8: Clr

In tal modo, infatti, si ripristinano i puntatori standard del Basic. Non consigliamo, a questo punto, di listare il programma nè, tantomeno, di modificarlo! I codici da noi inseriti, presenti da 2049, non hanno alcun significato per l'interprete, e correte il rischio di dover spegnere il computer.

Provvedete, invece, a registrare il programma con un banale:

Save "programma",8

...oppure...

Save "programma"

...se non disponete del drive, ma solo del registratore.

Con la modifica della locazione 44, e con la successiva operazione di registrazione, non abbiamo fatto altro che riportare, su supporto magnetico, "tutto" ciò che è presente in memoria Ram dalla locazione 2049 fino al valore puntato dalla coppia di locazioni 45 e 46; queste, dal momento che non sono state modificate, coincidono con la fine del nostro listato Basic.

• spegnete e riaccendete il C/64
• digitate:

Poke 44,10: Poke 2560,0: New

ripetendo, in partica, l'ordine visto in precedenza.

• caricate il programma di prima con il suffisso ".1" che obbliga il computer, come è noto, ad allocare il programma in oggetto nelle stesse identiche locazioni in cui si trovava al momento della registrazione:

Load "programma",8,1

Se dimenticate il suffisso ",1" sarete costretti a spegnere e ricominciare daccapo.

• Digitate:

Run 180

il messaggio, memorizzato nella prima fase, verrà ora visualizzato, dimostrando, in tal modo, che lo stesso messaggio è stato memorizzato insieme al programma al momento della registrazione.

• Provate ad aggiungere righe, modificarle, cancellarle: tutto funzionerà come di consueto ed un Run 180 darà sempre gli stessi risultati.

E' intuitivo che è possibile allocare routine in linguaggio macchina nella zona di memoria precedente al Basic "ufficiale"; la loro lunghezza potrà essere qualunque, a patto di inserire, nella locazione 44, il giusto valore.

Va da sè che, nei successivi caricamenti, è doveroso ricordarsi di alterare la locazione 44 (ed eventualmente la 43), e di annullare il byte precedente la locazione puntata; ciò va eseguito PRIMA di provvedere al caricamento stesso.

Analogamente, al momento delle registrazione, è di vitale importanza ripristinare il valore standard della locazione 44.

Due programmi in memoria

Sofisticando la tecnica descritta è possibile avere in memoria due (o più) programmi Basic, ognuno indipendente dall'altro. Provate a seguirci nell'interessante esperimento:

• spegnete e riaccendete il C/64.

• digitate con la massima attenzione (soprattutto la riga 130) il seguente mini-programma:

```
100 input"digita tre numeri": x, y, z
110 print"i numeri  sono:" x; y; z
120 print"la stringa è:" a$
130 poke 44,10: poke2560,0: goto 100
```

Non appena darete il Run, verrà chiesto di digitare tre numeri; rispondete digitando tre valori qualunque, separati da una virgola.

Subito dopo (riga 110) verrà data conferma dell'avvenuta memorizzazione dei tre numeri e comparirà un messaggio apparentemente strano ("La stringa è:") relativo ad una stringa (A$) nulla, dal momento che abbiamo impartito un Run.

Ma ciò che è più grave è sicuramente la presenza del messaggio "Undef'd Statment Error in 130".

A questo punto non tentate nemmeno lontanamente di chiedere un List, ma provvedete ad impartire il comando: New.
* Digitate, ora, il seguente listato:

```
100 input "digita la stringa";a$
110 print "i numeri  sono:"x ;y ;z
120 poke 44,8: goto 100
```

...e solo alla fine, dopo aver verificato la corretta trascrizione, date il Run. Possiamo, ora, capire ciò che accade.

Il primo programma non fa altro che chiedere tre numeri e associarli alle variabili X, Y e Z; subito dopo stampa il contenuto di A$ (che, per ora, non c'è).

L'ultima riga del primo listato impone la modifica del puntatore di inizio Basic (per semplicità non è stato considerato anche 43, ma i più pignoli, volendo...).

Il Goto 100 (vedi riga 130), nonostante l'impostazione della nuova "mappa" della memoria, "deve" essere interpretato, perchè rappresenta l'ultima istruzione della riga Basic attivata. In questo momento, però, l'interprete Basic è indotto a rintracciare la riga 100 e si prepara a farlo *considerando l'area di memoria ram in cui ritiene "esistente" il programma basic!* Questa, però, non è più la stessa di prima (a causa del Poke 44,10) e ne consegue la visualizzazione del messaggio di errore.

Quando, poi, digitate il secondo listato, questo viene allocato a partire da 2561 e attivato, alla fine, con un semplice Run.

In questo caso, però, giunto ad elaborare la riga 120 (dopo a-

ver spostato la mappa della memoria al valore originario), trova davvero "una" riga numerata con 100, ma non è più quella del secondo programma, bensì quella del primo.

Quest'ultimo, infine, non troverà più contraddizione giunto alla riga 130 perchè, stavolta, la riga 100 (pur se del secondo programma) "esiste" realmente.

La corretta visualizzazione delle quattro variabili interessate (X, Y, Z, A$) dimostra che, nei passaggi da un banco di memoria ad un altro, queste non vengono affette da alcuna variazione.

Il discorso seguito finora, come è intuitivo, può essere applicato a tre, quattro e, almeno in teoria, a un numero qualsiasi di programmi Basic, mutuamente richiamabili tra loro.

In pratica, però, il discorso si complica terribilmente: non meravigliatevi, quindi, se durante i vostri tentativi il computer si blocca senza rimedio.

Allo stesso modo una minima disattenzione nel registrare (o caricare) programmi scritti nel modo suggeriro può provocare vere catastrofi.

Il lettore, ad ogni modo, non tema di danneggiare il proprio calcolatore: nessuna operazione di Poke, per quanto ardita sia, riuscirà mai a provocar danni (se non la perdita di tempo nel digitare nuovamente i vari programmi).

E' superfluo ricordare che l'imposizione di Poke 44,10 è valida a patto di accontentarsi, per il primo programma, di soli 512 byte; per programmi più lunghi agire di conseguenza sulla locazione 44.

Sbatti il programma sullo schermo

Ciò che ora faremo servirà per meglio comprendere non solo il modo di operare dell'interprete Basic all'interno dell'area ad esso destinata, ma anche per suggerire nuove, inedite procedure software che possano costituire una valida base per intraprendere, tra l'altro, lo studio di inediti sistemi di protezione.

Per meglio comprendere l'argomento, e per consentire anche ai principianti di seguire il discorso, saremo costretti a riprendere argomenti forse già noti, sui quali, peraltro, promettiamo di intrattenerci il minimo indispensabile.

Una premessa

L'area di schermo è costituita da una successione di 1000 byte (25 righe * 40 colonne, caso del C/64), il cui indirizzo iniziale è 1024.

E' possibile scrivere caratteri sullo schermo non solo con istruzioni del tipo Print, ma anche con Poke e, in seguito, leggerne i codici con istruzioni Peek.

• Spegnete e riaccendete il C/64.
• Cancellate lo schermo (Shift + Clr/Home).
• Premete tre volte il tasto Return (il cursore, quindi, dovrebbe lampeggiare sul terzo rigo).
• Digitate: Poke 1024,1
• Nella prima cella in alto a sinistra dovrebbe apparire il carattere maiuscolo "A". Se ciò non avviene significa che il vostro C/64 è un modello dotato di una versione particolare di Rom che necessita della colorazione della cella video. Provate, in questo caso, con:

Poke 1024,0: Poke 55296,1

La memoria colore, come è noto, occupa le 1000 celle numerate da 55296 a 56295.
• Cancellate lo schermo e, rimanendo sulla prima cella video in alto a sinistra, battete il tasto "B" e scendete (SENZA premere il tasto Return, ma ricorrendo ai tasti cursore) di qualche riga, in modo da poter digitare liberamente:

Print Peek(1024)

• Otterete il valore 2. Ciò significa che ad ogni cella video è possibile associare uno dei 256 valori (numerati da 0 a 255) e che, viceversa, ad ogni carattere visualizzabile corrisponde uno dei 256 valori possibili.

Prima di proseguire ricordiamo che è possibile passare dal set maiuscolo-grafico a quello maiuscolo-minuscolo premendo i tasti Commodore e Shift. Lo stesso effetto si può ottenere con:

Print Chr$(14)

...per passare da maiuscolo-grafico a maiuscolo-minuscolo e con...

Print Chr$(142)

...per ritornare al maiuscolo-grafico.

Per fare in modo da rendere inefficiente la pressione contemporanea dei tasti Commodore e Shift, digitate:

Print Chr$(14); Chr$(8)

In questo modo non solo si imposta il set minuscolo, ma si impedisce di ritornare all'altro set premendo inavvertitamente i due tasti Shift e Commodore, a meno di premere Run/Stop e Restore, oppure di battere:

Print Chr$(9)

Alteriamo i puntatori

Se la vostra mente non è labile, dovreste ricordare che è possibile "spostare" l'area di memoria dedicata al Basic, limitandosi ad alterare i suoi puntatori di inizio (e magari, come vedremo tra breve, di fine).

Poichè l'area di schermo è una zona Ram come qualsiasi altra zona, faremo in modo che l'interprete Basic consideri proprio l'area video come zona in cui allocare eventuali programmi Basic!

Da questo momento è indispensabile seguire alla lettera, più che mai, le note che esporremo: un MINIMO errore vi costringerà a spegnere, riaccendere e ricominciare daccapo.

• Spegnete e riaccendete il C/64.
• Digitate...:

Print Chr$(14); Chr$(8)

...e cancellate lo schermo (Shift + Clr/Home).
• Premete dieci volte il tasto Return (non una di più, non una di meno).
• Battete, su una sola riga, i seguenti comandi:

Poke 44,4: Poke 1024,0: New

Chi, invece, dispone di un C/64 da "colorare", dovrà digitare alcuni comandi in più:

For i=55296 to 56295: Poke I,1: Next: Poke 44,4: Poke 1024,0: New

Immediatamente compariranno, in alto a sinistra, tre caratteri di "chiocciolina" (simbolo corrispondente al tasto situato tra l'asterisco e "P"). Che cosa è successo?
Il primo comando (Poke 44,4) sposta il Basic a partire da 4*256, cioè da 1024 che coincide, come abbiamo visto prima, proprio con l'inizio dell'area video.
Il successivo Poke 1024,0 è la "solita" locazione nulla che deve precedere l'inizio del Basic.
L'ultimo comando (New) ci evita di mettere a posto manualmente i rimamenti puntatori e, in pratica, obbliga il computer a sistemare tre zeri in successione all'inizio dell'area destinata al Basic (che coincide, lo ripetiamo, con l'area video).
Inutile dire che la chiocciolina rappresenta il codice zero nella rappresentazione delle Poke dello schermo!
Da questo momento è assolutamente necessario rendersi conto che non possiamo pasticciare sul video come siamo abituati di solito: dobbiamo tener presente, che ora, ad esempio, la cancellazione del video coincide con una cancellazione ben più grave.
Per digitare i vari comandi che suggeriremo tra breve, saremo costretti ad utilizzare sempre la stessa riga (la decima) per evitare disastrosi scrollig del video o, peggio, la comparsa di messaggi di errore che "sporcherebbero" lo schermo (e quindi, involontariamente, l'area del Basic).
In altre parole, tutti i comandi che indicheremo, dovrete digitarli sulla decima riga di schermo, armandosi di pazienza e salendo con il cursore su tale linea, cancellando i comandi precedenti (mediante la barra spaziatrice o il tasto Del) e, soprattutto, stando bene attenti ad evitare errori di battitura.
Per assicurarsi di essere sulla decima riga di schermo potete, fortunatamente, premere il tasto Home (e non Shift + Home) e premere dieci volte il tasto cursore in basso.
Abbiamo, insomma, attribuito alle 1000 locazioni di schermo l'ingrato compito di fungere, contemporaneamente, sia da area video che da area Basic, con la conseguenza di dover ri-

spettare tutte le esigenze di una doppia, delicata e, soprattutto, pesante responsabilità.

Le variabili

Abbiamo detto che è necessario stare attenti a non cancellare lo schermo per evitare guai. In effetti possiamo ripristinare le condizioni iniziali cancellando lo schermo con i tasti Shift e Clr/Home e digitando in successione, nelle prime tre celle video, altrettanti caratteri di chiocciolina.

Tale modo di comportarsi corrisponderà alla cancellazione del programma Basic eventualmente presente in memoria e avrà (quasi) lo stesso effetto di un New.

Supponendo, dunque, di aver impostato il modo maiuscolo-minuscolo, di aver spostato l'area Basic a partire dall'inizio del video, di avere lo schermo "pulito" (tranne le tre chioccioline in alto a sinistra) e di veder lampeggiare il cursore sulla decima riga, digitate il seguente comando:

AA=0

che associa, alla variabile in virgola mobile "AA", il valore nullo.

Vedrete subito una piccola rivoluzione in alto a sinistra dello stesso schermo:

- le tre chioccioline sono rimaste allo stesso posto (non vi sono ancora, infatti, programmi Basic).
- Subito dopo si notano i due caratteri "AA" che stanno a rappresentare il nome della prima varabile dichiarata (vedi inserto del N.43) che è anche l'unica.
- Dopo "AA" sono presenti cinque locazioni contenenti altrettante chioccioline che rappresentano il valore nullo. Queste cambieranno a seconda del valore successivamente associato ad "AA".

Salite, ora, con il cursore sulla riga che contiene AA=0 e modificatene il valore a piacimento (AA=123 AA=2.45 AA=-123.67): dopo ogni pressione del tasto Return noterete l'immediata modifica delle cinque locazioni poste dopo "AA" in alto sul video. Ciò è perfettamente in linea con quanto studiato nel primo inserto.

Naturalmente, ma agendo con la massima prudenza, potete posizionarvi, agendo con i tasti cursore, alla destra di "AA" (primo rigo del video), e digitare, a casaccio, vari caratteri (magari anche in reverse).

In seguito, sempre con i tasti cursore, posizionatevi sulla solita decima riga e impartite Print AA per verificare il valore che corrisponde a ciò che avete battuto.

Per esempio, se dopo AA (ci riferiamo ancora al primo rigo di schermo), battete "aaaaa" (cinque "a" minuscole in successione) otterrete, come risposta a Print AA, il valore 5.9696674e-38. Con "bbbbb", invece, il valore: 5.9696674e-39; e così via.

Non nominare invano le variabili

Facciamo, ora, una nuova esperienza:

• Cancellate lo schermo (Shift + Clr/Home)
• digitate tre chiocciolino in successione (equivale, come visto, a un New) e scendete, con i tasti cursore, sul decimo rigo.
• Battete il comando CLR, che annulla le variabili; tale comando è indispensabile. Provate, infatti, a farne a meno... (siete in grado di spiegarvi il motivo del successivo, strano comportamento?).
• digitate, sempre sul decimo rigo, il comando di prima: AA=0.
• Chiedete, ora, di stampare il contenuto della variabile BB:

Print BB

La risposta, ovviamente, sarà zero dal momento che non abbiamo dichiarato tale variabile in precedenza. Guardate in alto a sinistra: tutto è come prima.

Ma provate, ora, a commettere un errore di sintassi.

Digitate, ad esempio, BB e premete il tasto Return: oltre al Syntax Error (che ci aspettiamo), qualcosa è accaduto in alto, sullo schermo: subito dopo i sette byte relativi alla variabile "AA" sono comparsi altri sette byte relativi alla variabile ("ufficialmente" inesistente) "BB", che vale zero dal momento che vi sono cinque chiocciolino posizionate dopo i caratteri stessi "BB".

Questo fenomeno dimostra che, commettendo particolari errori di sintassi, costringiamo involontariamente l'interprete Basic a riservare spazio a variabili non richieste esplicitamente.

Con ciò si spiega, dunque, la presenza di variabili non dichiarate attivando particolari procedure Dump, presenti in Tool evoluti che aggiungono altri comandi a quelli standard Commodore.

Il primo programma

Vediamo, ora, che cosa accade digitando un "vero" programma Basic:

• Cancellate lo schermo e digitate in successione tre chiocciolinee.
• Scendete, con i tasti cursore, sul decimo rigo e digitate la seguente riga Basic:

255 rem commodore computer club

Sul primo rigo di schermo, non appena premiamo il tasto Return, vedremo il messaggio "Commodore Computer Club" (scritto tutto in maisucolo) a distanza di sette locazioni da sinistra e, subito dopo, le "solite" tre chiocciolinee (che indicano la fine del listato Basic).

Il primo carattere del video è una chiocciolina (zero) richiesta dall'interprete Basic. I successivi due byte sono i byte di Link alla prossima linea, e la coppia successiva rappresenta la numerazione Basic dell'unica riga digitata: si noti che, di questa coppia di byte, il primo è il carattere grafico 255 e il secondo la chiocciolina (0).

• Salite sulla riga Basic digitata sul decimo rigo di schermo
• Cancellatela con la barra spaziatrice
• Salite sul carattere semigrafico che rappresenta il valore 255 (due quadratini contrapposti).
• Sostituitelo con una chiocciolina.
• Posizionatevi nuovamente sul decimo rigo e chiedete il List: la numerazione non è più 255, ma zero (0).

In modo analogo, seguendo la falsariga di quanto fatto in precedenza, divertitevi a salire sulla prima riga, a sostituire il messaggio "Commodore Computer Club" con altri (facendo attenzione a non invadere gli altri byte) e a chiedere il List: ne vedrete delle belle; in alcuni casi, addirittura, sarete costretti a spegnere e riaccendere il computer.

Cancellate lo schermo, digitate le tre chiocciolìne e, posizionati sul decimo rigo, battete, dopo un opportuno New, il seguente programma:

0 For AA=1 to 10: For BB=1 to 10: For CC=1 to 10: Next CC,BB,AA

Non appena premerete il tasto Return, la prima riga di schermo, e qualche carattere della seconda, saranno riempiti da un apparente guazzabuglio di caratteri; lasciamo al lettore il compito di individuare i codici relativi ai comandi, alle variabili, al termine della riga e così via.
Battete Run e osservate lo schermo: subito dopo le tre chiocciolìne di fine Basic compaiono, all'improvviso, altri 21 caratteri che rappresentano, a gruppi di sette, le tre variabili "AA" "BB" "CC" dichiarate dallo stesso programma; non solo, ma noterete che, grazie ai tre cicli For...Next nidificati tra loro, il valore (e quindi i corrispondenti caratteri visualizzati sullo schermo) cambia ad ogni ciclo. Il risultato, insomma, consiste nell'osservare comodamente, sullo schermo, ciò che avviene, in tempo reale, all'interno del computer quando vengono elaborate variabili numeriche.

Il Top di Memoria

Prima di studiare ciò che accade alle stringhe, consigliamo caldamente di rileggere le notizie che le riguardano (inserto N.43).
E' bene ricordare che l'ultima locazione utilizzabile dall'interprete Basic è puntata dalla coppia 55 e 56.
Per sincerarcene effettuiamo un semplice esperimento:

- Spegnete e riaccendete il computer.

• Cancellate lo schermo, "scendete" sulla *ventesima* riga con i tasti cursore e digitate il seguente gruppo di comandi:

Print Chr$(14);Chr$(8): Poke 44,4: Poke 1024,0: Poke 56,6: New

In tal modo non solo facciamo coincidere l'inizio del Basic con l'inizio dall'area di schermo (come nelle esperienze di prima), ma limitiamo il cosiddetto "Top di memoria" alla locazione 6*256=1536.

Se, infatti, salite con il cursore sulla ventesima riga, la cancellate con la barra spaziatrice e chidete un banale:

Print Fre (0)

la risposta laconica è un modesto 509.

Risalite, con pazienza, sul ventesimo rigo, cancellate ciò che eventualmente rimane e battete:

AA$="commodore computer club"

Al momento della pressione del tasto Return notiamo che alle tre chiocciolìne presenti in alto sul video si affiancano i sette byte che rappresentano le peculiatrità della variabile stringa appena dichiarata.

Ma ciò che più attira la nostra attenzione è il fatto che il messaggio "Commodore Computer Club" compare nel bel mezzo dello schermo; i più pignoli potranno verificare che l'ultimo carattere del messaggio coincide con la cella video 1535, che precede immediatamente la cella 1536 prima calcolata.

Le sorprese non sono finite: se saliamo ancora con il cursore sulla riga che contiene...

AA$="commodore computer club"

...e battiamo nuovamente il tasto Return, ci accorgiamo che un nuovo messaggio viene visualizzato nelle locazioni di memoria che precedono quello già visibile.

Ne deduciamo che:

• Ogni volta che si dichiara una variabile stringa, il suo contenuto "sale" all'interno della memoria Ram disponibile per

l'interprete Basic, fino a giungere all'ultima locazione che contiene l'ultima variabile dichiarata; a partire da questo istante eventuali altre variabili stringa ricominciano il ciclo a partire dal "fondo" disponibile a meno che non si impartisca un Run, un Clr oppure si richieda un Fre(0).

Supponendo di NON aver spento e riacceso il computer, cancellate lo schermo, digitate le familiari tre chiocciolinc, scendete sul ventesimo rigo e impartite un opportuno New.

Subito dopo, risaliti sul ventesimo rigo, battete il microprogramma che segue:

100 For II=1 to 100: AA$=""+"pippo": For JJ=1 to 300: Next JJ,II

Se non ricorressimo al trucco della concatenazione delle stringhe, infatti (vedi N.43), non potremmo osservare il fenomeno della "risalita" delle stringhe, visibile con il solito Run.

Impartendo il comando Run, infatti, vedremo, come prima, la manipolazione in tempo reale della variabile II (che varia tra 1 e 100), di JJ (tra 1 e 300) e di AA$, in cui cambiano continuamente i puntatori che individuano, istante dopo istante, la posizione esatta, all'interno della Ram, della stringa elaborata.

L'esperienza effettuata consente di meglio comprendere la necessità di impostare un Top di memoria nel caso in cui si desideri allocare programmi in linguaggio macchina (o messaggi) in una zona di memoria a torto ritenuta libera: in caso contrario, infatti, la "salita" delle stringhe cancella inesorabilmente qualsiasi "cosa" incontri nel suo distruttivo cammino.

...E altre idee

Con queste ultime considerazioni termina la camminata (è il caso di dirlo) all'interno della Ram gestita dall'interprete Basic.

Ci limiteremo a sollecitare i lettori ad esplorare nuovi sistemi di protezione che utilizzino lo spostamento dell'area destinata al Basic, spostandola magari nello schermo, durante la registrazione, e il caricamento, di programmi.

Come al solito i migliori lavori che dovessero pervenire in Redazione saranno adeguatamente compensati.

Vale, ovviamente, il solito consiglio di telefonare (tel. 02/8467348) prima di inviare il frutto del vostro lavoro.